

25252/H/06



PEMBUATAN APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA DI KOTA MOJOKERTO

TUGAS AKHIR

RSSI

005.74
Suk

p-1
2006



Disusun Oleh :

PERPUSTAKAAN ITS	
Tgl. Terima	21 - 2 - 06
Terima Dari	H
No. Agenda Prp.	224283

Yan Andriariza Ambhita Sukma

NRP. 5201 100 033

PEMBUATAN APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA DI KOTA MOJOKERTO

TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Komputer**

Pada

Program Studi Sistem Informasi

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui/Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II

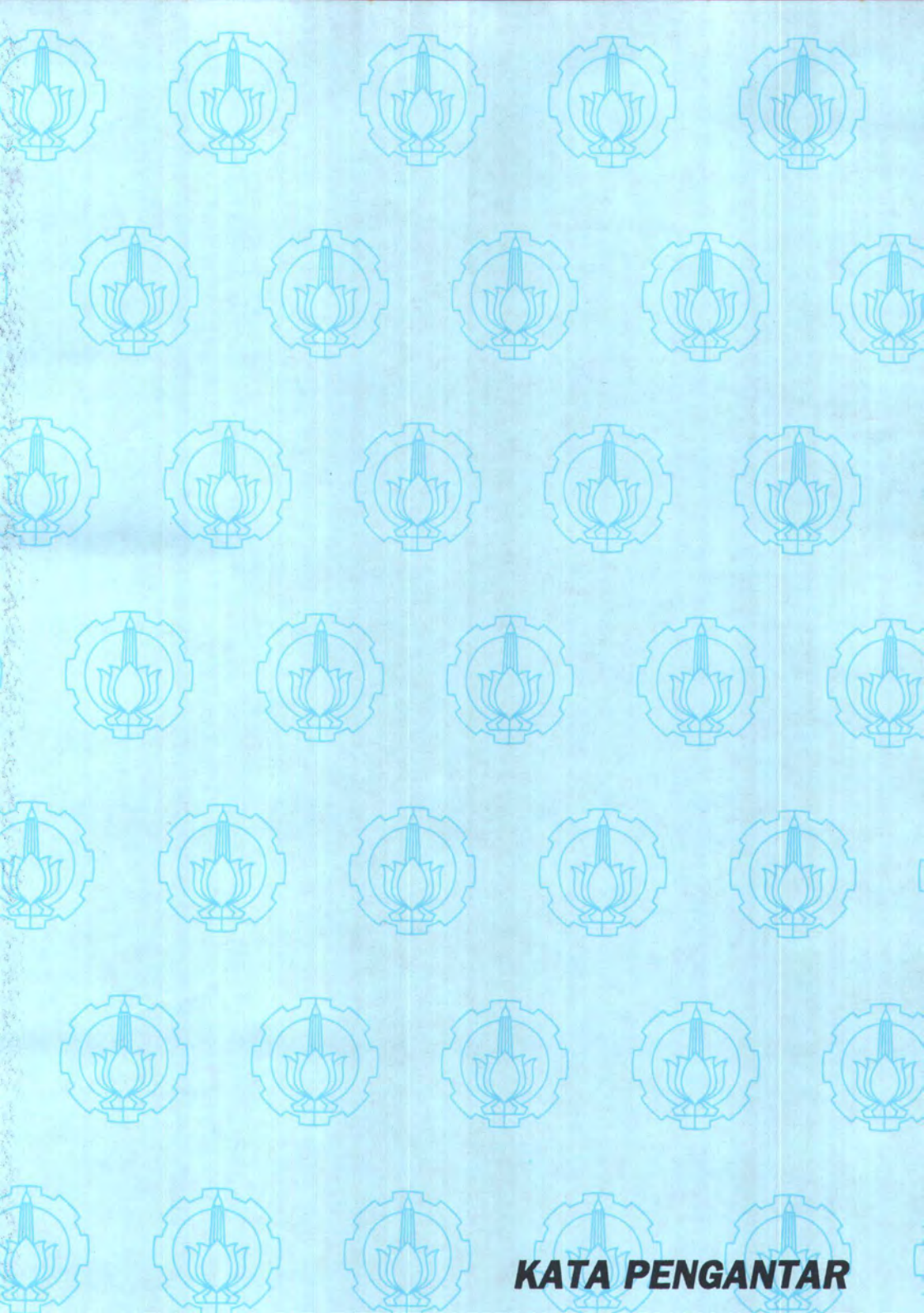


ABSTRAK

Perencanaan trayek angkutan kota di Pemkot Mojokerto yang dilakukan selama ini prosesnya masih belum terintegrasi. Sebagian prosesnya masih dikerjakan secara manual dan sebagian lainnya sudah menggunakan aplikasi pendukung, bukan aplikasi yang khusus digunakan untuk perencanaan trayek angkutan kota. Karena itulah diperlukan sebuah aplikasi khusus untuk perencanaan trayek angkutan kota yang prosesnya sudah terintegrasi dalam sebuah aplikasi saja, sehingga pengguna hanya perlu memasukkan data yang dibutuhkan dan langsung memperoleh hasil akhirnya, tanpa perlu melakukan proses secara terpisah-pisah.

Tugas akhir ini berkaitan dengan pembuatan aplikasi untuk perencanaan trayek angkutan kota yang didasarkan pada ilmu transportasi. Perencanaan yang dimaksud adalah merencanakan kembali rute trayek angkutan kota yang sudah ada di Kota Mojokerto. Aplikasi ini menerima masukan berupa data kependudukan, data wilayah, data jaringan jalan, data trayek, dan data survei *Home Interview* yang dilakukan oleh Dinas Perhubungan Kota Mojokerto. Semua masukan tersebut kemudian diproses dengan menggunakan "Model Perencanaan Transportasi Empat Tahap", yang meliputi pembangkitan perjalanan, persebaran perjalanan, pemilihan rute dan pemilihan moda. Dari proses tersebut kemudian diperoleh luaran berupa rute baru beserta hasil evaluasinya, yaitu nilai rasio waktu perjalanan, rasio biaya perjalanan, persentase penggunaan angkutan kota, serta pangsa pasar untuk rute lama dan rute baru pada tahun sekarang dan tahun rencana. Dari hasil evaluasi tersebut dapat diketahui rute mana yang sebaiknya dipilih untuk masing-masing trayek yang ada.

Aplikasi yang telah dibuat telah dilakukan uji coba dengan menggunakan data survei tahun ini untuk melakukan perencanaan tahun 2006. Dari hasil uji coba yang dilakukan, aplikasi ini telah berhasil memperoleh luaran berupa rute baru masing-masing trayek beserta evaluasinya dan saran penggunaan rute untuk masing-masing trayek tersebut. Untuk ini terdapat 2 trayek yang mengalami perubahan rute menjadi lebih cepat untuk rute barunya dibandingkan dengan rute lamanya, dan berdasarkan hasil evaluasi rasio waktu perjalanan, rasio biaya perjalanan, persentase penggunaan trayek dan pangsa pasarnya, maka kedua trayek tersebut disarankan untuk menggunakan rute barunya.



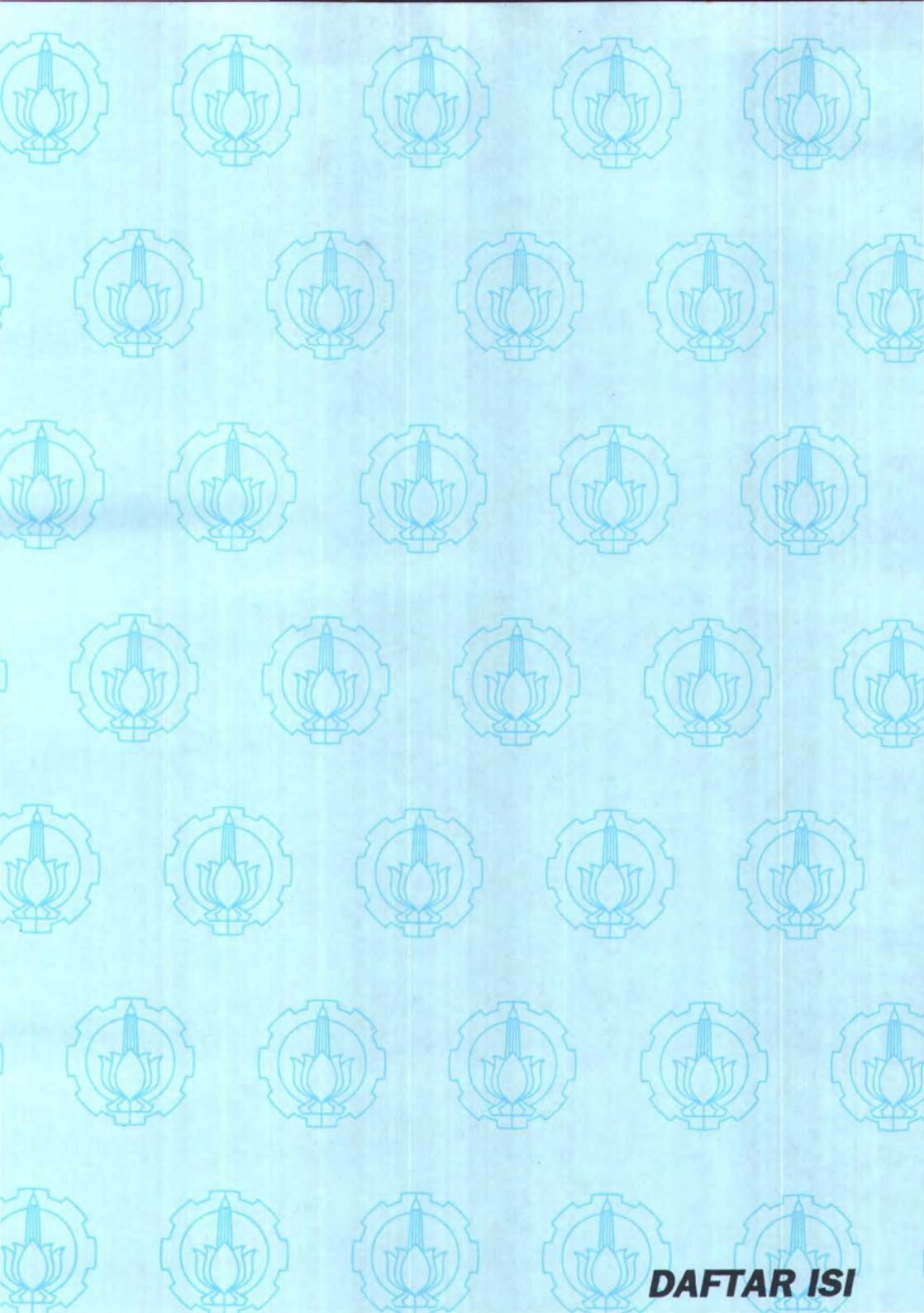
KATA PENGANTAR

KATA PENGANTAR

Segala puji syukur yang tiada akan pernah terhenti kehadiran Allah SWT atas segala limpahan hidayah, pertolongan dan kasih sayang-Nya sehingga akhirnya penulis dapat menyelesaikan Tugas Akhir yang berjudul **PEMBUATAN APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA DI KOTA MOJOKERTO** yang merupakan salah satu syarat kelulusan strata satu (S1) Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh November Surabaya.

Tugas akhir ini tidak akan pernah terwujud tanpa bantuan, saran dan bimbingan dari berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Tugas Akhir ini khusus penulis persembahkan untuk kedua orang tua penulis. Terima kasih atas bimbingan, nasehat, doa, kasih sayang, perhatian, dukungan moril dan spirituil, serta semua yang selalu diberikan tiada henti-hentinya pada penulis. Semoga penulis dapat membalas dan semoga kedua orang tua penulis selalu diberi perlindungan, limpahan rahmat dan kasih sayang yang tiada henti-hentinya pula dari Allah SWT. Amin.
2. Kepada keluarga besar penulis, mbak ira, mas febby, mbak nurin, mas anto, dik azmi, dik rizky. Makasih buat semangat, dorongan, doa, telepon yang menguatkan hati dan nasehatnya. Makasih!!!!
3. Bapak Arif Djunaidy selaku pembimbing pertama penulis, terima kasih atas bimbingan yang diberikan selama ini kepada penulis.



DAFTAR ISI

ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR PROGRAM	xix
BAB I PENDAHULUAN	1
1. LATAR BELAKANG.....	1
2. PERMASALAHAN	2
3. BATASAN MASALAH	2
4. TUJUAN	5
5. MANFAAT	5
6. METODOLOGI PENULISAN TUGAS AKHIR.....	5
7. SISTEMATIKA PENULISAN	6
BAB II DASAR TEORI	8
1. PERAMALAN PERMINTAAN PERJALANAN	8
2. 1. 1. Tahap Pembangkitan Perjalanan	8
2. 1. 1. 1. Analisis Regresi Linier.....	9
2. 1. 1. 2. Analisis Regresi Linier Berganda.....	11
2. 1. 1. 3. Uji Statistika.....	17
2. 1. 1. 4. Metode Analisis Langkah Demi Langkah Tipe 1	21
2. 1. 1. 5. Rumus Tingkat Pertumbuhan.....	23
2. 1. 2. Tahapan Distribusi Perjalanan	23
2. 1. 3. Tahapan Pemilihan Moda.....	31
2. 1. 3. 1. Tahapan I.....	34
2. 1. 3. 2. Tahapan III	37
2. 1. 4. Tahapan Pemilihan Rute	39
2. PERLUASAN DATA	41

3. 2. PERANCANGAN DATA	54
3. 2. 1. Pembuatan ERD	55
3. 2. 1. 1. ERD Data Survey	55
3. 2. 1. 2. ERD Data Trayek	61
3. 2. 1. 3. ERD Data Kependudukan	62
3. 2. 2. Pemetaan ERD Menjadi Tabel	66
3. 2. 3. Normalisasi Data	67
3. 2. 3. 1. Ketergantungan Fungsional	67
3. 2. 3. 2. Normalisasi 2 NF	70
3. 2. 3. 3. Normalisasi 3 NF	70
3. 2. 3. 4. Tabel Setelah Normalisasi	70
3. 3. PERANCANGAN PROSES	73
3. 3. 1. DFD Level 0 (Diagram Kontek Data)	73
3. 3. 2. Aplikasi Perencanaan Trayek Angkutan Kota	74
3. 3. 2. 1. Masukan Basis-Data	74
3. 3. 2. 1. 1. Proses Masukan Data Kependudukan	76
3. 3. 2. 1. 2. Proses Masukan Data Trayek	80
3. 3. 2. 1. 3. Proses masukan data survey HI	81
3. 3. 2. 1. 4. Proses masukan data wilayah	85
3. 3. 2. 2. Proses Perencanaan Trayek	91
3. 3. 2. 2. 1. Peramalan Matrik Asal Tujuan Perjalanan	94
3. 3. 2. 2. 2. Pencarian Rute Tercepat Antar Zona	103
3. 3. 2. 2. 3. Pencarian Jalur Baru Berdasarkan Rute Alternatif Tercepat Dan Rute Lama	107
3. 3. 2. 2. 4. Evaluasi RWP dan RBP Rute Baru Terhadap Kendaraan Pribadi	107
3. 3. 2. 2. 5. Persentase Dan Pangsa Pasar Penggunaan Angkutan	107
3. 4. PERANCANGAN ANTAR MUKA	112

BAB IV PEMBUATAN APLIKASI 116

1. 1. PEMBUATAN BASIS DATA	116
1. 2. PEMBUATAN PROSES APLIKASI	124
4. 2. 1. Masukan Basis-Data	124
4. 2. 1. 1. Proses Masukan Data Kependudukan	124
4. 2. 1. 1. 1. Proses Masukan Data Jumlah Penduduk	124

4. 2. 1. 4. 3.	Masukan Data Kelurahan	130
4. 2. 1. 4. 4.	Masukan Data Wilayah Zona	131
4. 2. 1. 4. 5.	Masukan Data Link	132
4. 2. 1. 4. 6.	Masukan Data Node	134
4. 2. 2.	Proses Perencanaan Trayek	136
4. 2. 2. 1.	Peramalan Matrik Asal Tujuan Perjalanan.....	136
4. 2. 2. 1. 1.	Pencarian Persamaan Regresi.....	136
4. 2. 2. 1. 2.	Peramalan Data Kependudukan dan Bangkitan Perjalanan.....	147
4. 2. 2. 1. 3.	Perluasan Data Survey	149
4. 2. 2. 1. 4.	Peramalan Perjalanan Antar Zona.....	151
4. 2. 2. 2.	Proses Pencarian Rute Tercepat Antar Zona	151
4. 2. 2. 3.	Proses Pencarian Jalur Baru Berdasarkan Rute Alternatif Tercepat Dan Rute Lama.....	153
4. 2. 2. 4.	Proses Evaluasi RWP dan RBP Rute Baru Terhadap kendaraan Pribadi	158
4. 2. 2. 5.	Proses Persentase Dan Pangsa Pasar Penggunaan Angkutan.....	159
4. 2. 2. 6.	Unit Dijkstra.....	160
4. 2. 2. 6. 1.	Fungsi GetRecNo.....	160
4. 2. 2. 6. 2.	Fungsi DirectPathCost1	164
4. 2. 2. 6. 3.	Fungsi DirectPathCost2	164
4. 2. 2. 6. 4.	Fungsi Dijkstra.....	166
3.	PEMBUATAN ANTARMUKA APLIKASI	169
4. 3. 1.	Home (Tampilan Utama).....	169
4. 3. 2.	Antarmuka Untuk Masukan Data (Tampilan Masukan)	170
4. 3. 2. 1.	Tampilan Masukan Data Kependudukan	170
4. 3. 2. 2.	Tampilan Masukan Trayek.....	171
4. 3. 2. 3.	Tampilan Masukan Survey HI	172
4. 3. 2. 4.	Tampilan Masukan Data Kewilayahan	173
4. 3. 3.	Tampilan Proses Perencanaan Ulang Trayek Angkutan	175
4. 3. 3. 1.	Tampilan Masukan Tahun Untuk Proses Perencanaan Ulang Trayek Angkutan	175
4. 3. 3. 2.	Hasil Laporan Proses Perencanaan Ulang Trayek Angkutan.....	176

5. 4. 1. 2.	Hasil Uji Coba Skenario 1	179
5. 4. 1. 3.	Analisis Hasil Skenario 1	180
5. 4. 2.	Pelaksanaan Uji Coba Dan Analisis Hasil Faktor Ekspansi (Skenario 2).....	180
5. 4. 2. 1.	Langkah Pelaksanaan Uji Coba Skenario 2	181
5. 4. 2. 2.	Hasil Uji Coba Skenario 2.....	181
5. 4. 2. 3.	Analisis Hasil Skenario 2	182
5. 4. 3.	Pelaksanaan Uji Coba Dan Analisis Hasil Mencari Faktor Pertumbuhan Dan Peramalan Data Kependudukan (Skenario 3)	182
5. 4. 3. 1.	Langkah Pelaksanaan Uji Coba Skenario 3	183
5. 4. 3. 2.	Hasil Uji Coba Skenario 3.....	183
5. 4. 3. 3.	Analisis Hasil Skenario 3	186
5. 4. 4.	Pelaksanaan Uji Coba Dan Analisis Hasil Metode Furnees (Skenario 4).....	186
5. 4. 4. 1.	Langkah Pelaksanaan Uji Coba Skenario 4	186
5. 4. 4. 2.	Hasil Uji Coba Skenario 4.....	187
5. 4. 4. 3.	Analisis Hasil Skenario 4	187
5. 4. 5.	Pelaksanaan Uji Coba Dan Analisis Hasil Algoritma <i>Dijkstra</i> (Skenario 5).....	188
5. 4. 5. 1.	Langkah Pelaksanaan Uji Coba Skenario 5	188
5. 4. 5. 2.	Hasil Uji Coba Skenario 5.....	188
5. 4. 5. 3.	Analisis Hasil Skenario 5	188
5. 4. 6.	Pelaksanaan Uji Coba Dan Analisis Hasil Pencarian Persamaan Regresi Persentase Penggunaan Angkutan (Skenario 6).....	189
5. 4. 6. 1.	Langkah Pelaksanaan Uji Coba Skenario 6	189
5. 4. 6. 2.	Hasil Uji Coba Skenario 6.....	189
5. 4. 6. 3.	Analisis Hasil Skenario 6	190
5. 5.	PELAKSANAAN UJI COBA DAN ANALISIS HASIL VALIDASI APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA (SKENARIO 7).....	191
5. 5. 1.	Langkah Pelaksanaan Uji Coba Skenario 7	191
5. 5. 2.	Hasil Uji Coba Skenario 7.....	192
5. 5. 3.	Analisis Hasil Skenario 7	195
BAB VI PENUTUP		201
5. 1.	SIMPULAN	201

DAFTAR TABEL

Tabel 3.1 Daftar Kecamatan dan Desa di Kota Mojokerto (Sumber : Web Site Kota Mojokerto).....	47
Tabel 3.2 Rute Trayek Angkutan Kota di Kota Mojokerto (Laporan Umum PKL ^[11]).....	49
Tabel 3.3 Pembagian Wilayah Mojokerto Dalam Zona (Laporan Umum PKL)	50
Tabel 3.4 Perbandingan antara kondisi sistem saat ini dan yang diinginkan	53
Tabel 3.5 Deskripsi Tiap Entitas dan Atribut Pemodelan Data Survey	57
Tabel 3.6 Deskripsi Tiap Entitas dan Atribut Pemodelan Data Trayek	64
Tabel 3.7 Deskripsi Tiap Entitas dan Atribut Pemodelan Data Kependudukan	65
Tabel 3.8 Tabel Zona.....	70
Tabel 3.9 Tabel Kelurahan	70
Tabel 3.10 Tabel Wilayah Zona	71
Tabel 3.11 Tabel Node	71
Tabel 3.12 Tabel Jalan.....	71
Tabel 3.13 Tabel Link	71
Tabel 3.14 Tabel Asal tujuan Perjalanan.....	71
Tabel 3.15 Tabel Total Data Keluarga	72
Tabel 3.16 Tabel Sampel HI.....	72
Tabel 3.17 Tabel Trayek	72
Tabel 3.18 Tabel Rute	72
Tabel 3.19 Tabel Komponen Kependudukan.....	72
Tabel 3.20 Tabel Kependudukan	73

Tabel 5.5 Perhitungan Manual Untuk Mendapatkan Data Kependudukan Dan Perjalanan Pada Tahun Rencana	185
Tabel 5.6 Tabel Hasil Proses Aplikasi Untuk Meramalkan Data Kependudukan Dan Bangkitan Perjalanan Pada Suatu Tahun Rencana	185
Tabel 5.7 Data Awal Perjalanan Antar Zona.....	187
Tabel 5.8 Iterasi Keempat Perhitungan Perjalanan Antar Zona Untuk Suatu Tahun Rencana.....	187
Tabel 5.9 Tabel Perbandingan Persen Kesalahan Antara Data Asli Dengan Hasil Persamaan Regresi	190



DAFTAR GAMBAR

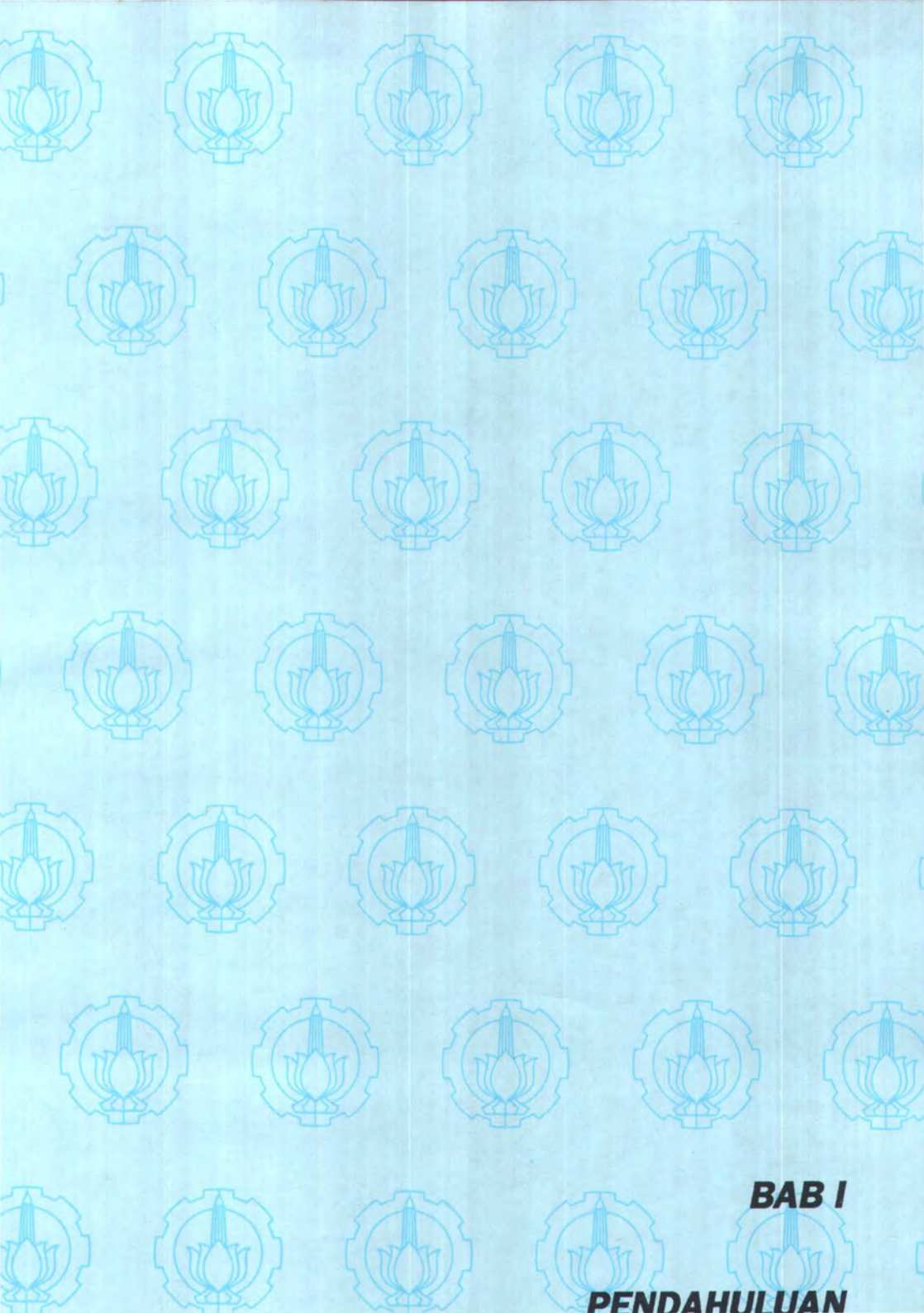
Gambar 2.1	Proses Kalibrasi dan Pengabsahan Model Analisis Regresi	17
Gambar 2.2	Metode untuk mendapatkan MAT (Sumber: O.Z. Tamin, Perencanaan Dan Pemodelan Transportasi).....	26
Gambar 2.3	Model Tahapan Perencanaan Transportasi.....	33
Gambar 2.4	Jaringan Trayek Pola <i>Radial</i>	42
Gambar 2.5	Jaringan Trayek Pola <i>Orthogonal/Grid</i>	43
Gambar 2.6	Jaringan Trayek Pola <i>Radial</i> Bersilang	43
Gambar 2.7	Jaringan Trayek Pola Jalur Utama dengan <i>Feeder</i>	44
Gambar 2.8	Jaringan Trayek Pola <i>Time Transfer Network</i>	44
Gambar 3.1	Peta Kota Mojokerto (Sumber: Website Kota Mojokerto)	48
Gambar 3.2	Pemodelan Data Survey	56
Gambar 3.3	Pemodelan data trayek	63
Gambar 3.4	Pemodelan Data Kependudukan	63
Gambar 3.5	Pemetaan ERD Data Survey	66
Gambar 3.6	Pemetaan ERD Data Trayek	67
Gambar 3.7	Pemetaan ERD Data Kependudukan.....	67
Gambar 3.8	Ketergantungan Fungsional Tabel Zona	68
Gambar 3.9	Ketergantungan Fungsional Tabel Kelurahan.....	68
Gambar 3.10	Ketergantungan Fungsional Tabel Wilayah Zona.....	68
Gambar 3.11	Ketergantungan Fungsional Tabel Node.....	68
Gambar 3.12	Ketergantungan Fungsional Tabel Link	68
Gambar 3.13	Ketergantungan Fungsional Tabel Jalan	69
Gambar 3.14	Ketergantungan Fungsional Tabel Total Data Keluarga	69

Gambar 3.21 Diagram Kontek Data Aplikasi Untuk Perencanaan Trayek Angkutan Kota	74
Gambar 3.22 DFD Level 1 Aplikasi Untuk Perencanaan Trayek Angkutan Kota	75
Gambar 3.23 DFD Level 2 Masukan Basis-Data.....	76
Gambar 3.24 DFD Level 3 Masukan Data Kependudukan.....	77
Gambar 3.25 <i>Flowchart</i> Proses Masukan Data Jumlah Penduduk (Proses 1.1.1)	78
Gambar 3.26 <i>Flowchart</i> Proses Masukan Data Pendapatan (Proses 1.1.2)	79
Gambar 3.27 <i>Flowchart</i> Proses Masukan Data Kendaraan (Proses 1.1.3)	80
Gambar 3.28 <i>Flowchart</i> Proses Masukan Data Trayek (Proses 1.2)	81
Gambar 3.29 DFD Level 3 Masukan Data Survey HI	82
Gambar 3.30 <i>Flowchart</i> Proses Masukan Survey HI (Proses 1.3.1).....	83
Gambar 3.31 <i>Flowchart</i> Proses Masukan Asal Tujuan Perjalanan (Proses 1.3.2)	84
Gambar 3.32 <i>Flowchart</i> Proses Masukan Sampel Survey (Proses 1.3.3).....	85
Gambar 3.33 DFD Level 3 Masukan Data Wilayah	86
Gambar 3.34 <i>Flowchart</i> Proses Masukan Data Zona (Proses 1.4.1)	87
Gambar 3.35 <i>Flowchart</i> Proses Masukan Data Jalan (Proses 1.4.2)	87
Gambar 3.36 <i>Flowchart</i> Proses Data Kelurahan (Proses 1.4.3).....	88
Gambar 3.37 <i>Flowchart</i> Proses Masukan Data Wilayah Zona (Proses 1.4.4).....	89
Gambar 3.38 <i>Flowchart</i> Proses Masukan Data Link (Proses 1.4.5)	90
Gambar 3.39 <i>Flowchart</i> Proses Masukan Data Node (Proses 1.4.6)	91
Gambar 3.40 Bagan Alir Perencanaan Trayek Angkutan Kota	93
Gambar 3.41 DFD Level 2 Perencanaan Trayek Angkutan Kota.....	94
Gambar 3.42 DFD Level 3 Peramalan Matrik Asal Tujuan Perjalanan.....	95
Gambar 3.43 <i>Flowchart</i> Proses Peramalan Data Kependudukan (Proses 2.1.1)	

Gambar 4.9 Masukan Data Ke <i>DataField</i> Total_Pendapatan dan <i>DataSource</i> DataModuleSPK.Data_Total_Data_Keluarga	127
Gambar 4.10 Masukan Data Ke <i>DataField</i> Total_Kendaraan dan <i>DataSource</i> DataModuleSPK.Data_Total_Data_Keluarga	128
Gambar 4.11 Masukan Data Ke <i>DataField</i> Zona_Asal dan <i>DataSource</i> DataModuleSPK.Data_Asal_Tujuan_Perjalanan.....	128
Gambar 4.12 Masukan Data Ke <i>DataField</i> Zona_Tujuan dan <i>DataSource</i> DataModuleSPK.Data_Asal_Tujuan_Perjalanan.....	128
Gambar 4.13 Masukan Data Ke <i>DataField</i> Jumlah_Perjalanan dan <i>DataSource</i> DataModuleSPK.Data_Asal_Tujuan_Perjalanan.....	128
Gambar 4.14 Masukan Data Ke <i>DataField</i> Jumlah_Perjalanan_Angkutan dan <i>DataSource</i> DataModuleSPK.Data_Asal_Tujuan_Perjalanan	128
Gambar 4.15 Masukan Data Ke <i>DataField</i> ID_Zona dan <i>DataSource</i> DataModuleSPK.Data_Sampel_HI.....	129
Gambar 4.16 Masukan Data Ke <i>DataField</i> Jumlah_Keluarga_Disurvey dan <i>DataSource</i> DataModuleSPK.Data_Sampel_HI.....	129
Gambar 4.17 Masukan Data Ke <i>DataField</i> Jumlah_Seluruh_Keluarga dan <i>DataSource</i> DataModuleSPK.Data_Sampel_HI.....	129
Gambar 4.18 Masukan Data Ke <i>DataField</i> Keluarga_Menolak_Diwawancara dan <i>DataSource</i> DataModuleSPK.Data_Sampel_HI.....	129
Gambar 4.19 Masukan Data Ke <i>DataField</i> Jumlah_Rumah_Kosong dan <i>DataSource</i> DataModuleSPK.Data_Sampel_HI.....	130
Gambar 4.20 Masukan Data Ke <i>DataField</i> Nama_Zona dan <i>DataSource</i> DataModuleSPK.Data_Zona.....	130
Gambar 4.21 Masukan Data Ke <i>DataField</i> Nama_Jalan dan <i>DataSource</i> DataModuleSPK.Data_Jalan.....	130
Gambar 4.22 Masukan Data Ke <i>DataField</i> Nama_Kelurahan dan <i>DataSource</i> DataModuleSPK.Data_Kelurahan	131
Gambar 4.23 Mengambil Data Nama_kelurahan Dengan KeyField ID_Kelurahan Dari <i>ListSource</i> DataModuleSPK.Data_Kelurahan	132

Gambar 4.28 Masukan Data Ke <i>DataField</i> ID_Node_Akhir dan <i>DataSource</i> <i>dataModuleSPK.Data_Link</i>	134
Gambar 4.29 Masukan Data Ke <i>DataField</i> Panjang dan <i>DataSource</i> <i>dataModuleSPK.Data_Link</i>	134
Gambar 4.30 Masukan Data Ke <i>DataField</i> Arus_Lalu_Lintas dan <i>DataSource</i> <i>dataModuleSPK.Data_Link</i>	134
Gambar 4.31 Masukan Data Ke <i>DataField</i> Kapasitas dan <i>DataSource</i> <i>dataModuleSPK.Data_Link</i>	134
Gambar 4.32 Masukan Data Ke <i>DataField</i> Kecepatan dan <i>DataSource</i> <i>dataModuleSPK.Data_Link</i>	134
Gambar 4.33 Tampilan Utama (<i>Home</i>).....	170
Gambar 4.34 Tampilan Untuk Masukan Jumlah Penduduk	171
Gambar 4.35 Tampilan Untuk Masukan Jumlah Pendapatan	171
Gambar 4.36 Tampilan Untuk Masukan Jumlah Kendaraan	171
Gambar 4.37 Tampilan Masukan Data Trayek	172
Gambar 4.38 Tampilan masukan jumlah total data survey HI.....	172
Gambar 4.39 Tampilan Masukan Data Asal Tujuan Perjalanan	173
Gambar 4.40 Tampilan Masukan Data Sampel HI	173
Gambar 4.41 Tampilan Masukan Data Zona	174
Gambar 4.42 Tampilan Masukan Data Kelurahan.....	174
Gambar 4.43 Tampilan Masukan Data Wilayah Zona.....	174
Gambar 4.44 Tampilan Masukan Data Node.....	174
Gambar 4.45 Tampilan Masukan Data Link	175
Gambar 4.46 Tampilan Masukan Data Jalan	175
Gambar 4.47 Tampilan Untuk Masukan Tahun Perencanaan.....	176
Gambar 4.48 Tampilan Dasar Hasil Proses Perencanaan	176
Gambar 5.1 Hasil Proses Mencari Persamaan Regresi Jumlah Perjalanan.....	179

ambar 5.8 Evaluasi RWP Dan RBP Masing-Masing Trayek.....	193
ambar 5.9 Evaluasi Persentase Penggunaan Angkutan Kota Masing-Masing trayek	194
ambar 5.10 Evaluasi Pangsa Pasar Angkutan Kota Masing-Masing Trayek.....	194
ambar 5.11 Hasil Akhir Penentuan Penggunaan Rute Trayek Angkutan Kota.....	195
ambar 5.12 Peta Rute Lama Dan Baru Trayek A Untuk Arah Berangkat	197
ambar 5.13 Peta Rute Lama Dan Baru Trayek A Arah Pulang.....	198
ambar 5.14 Peta Rute Lama Dan Baru Trayek G Untuk Arah Berangkat	199



BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

Pada bab ini dijelaskan beberapa hal dasar meliputi latar belakang, permasalahan, batasan permasalahan, tujuan, manfaat, metodologi dan sistematika penulisan. Diharapkan bab ini akan memberikan gambaran umum terhadap permasalahan dan pemecahannya yang dapat digunakan untuk menyelesaikan Tugas Akhir.

1. 1. LATAR BELAKANG

Angkutan kota merupakan salah satu bentuk dari angkutan umum yang mempunyai fungsi sebagai sarana pergerakan manusia untuk berpindah dari suatu tempat ke tempat lain. Sarana ini merupakan sarana transportasi alternatif di dalam kota, terutama bagi masyarakat yang tidak memiliki kendaraan pribadi. Sehingga kebutuhan akan sarana dan prasarana ini sangat diperlukan di wilayah perkotaan, dalam hal ini termasuk kota Mojokerto yang letaknya tidak jauh dari kota Surabaya.

Angkutan kota diharapkan mampu menyediakan *aksesibilitas* yang baik bagi penggunaanya, dimana hal ini dapat dilihat dari dua faktor yang menentukan tingkat tinggi rendahnya akses dari suatu tempat asal tujuan. Faktor tersebut adalah faktor waktu tempuh dan faktor biaya perjalanan. Dengan semakin kecilnya kedua faktor tersebut bila dibandingkan dengan penggunaan kendaraan pribadi, maka tingkat *aksesibilitas* dengan menggunakan angkutan kota menjadi semakin tinggi, sehingga diharapkan penggunaan moda kendaraan pribadi akan berkurang dan beralih ke moda angkutan kota.

- . Tahapan-tahapan yang digunakan dalam proses perencanaan transportasi adalah empat tahap, yaitu Tahap Pembangkitan Perjalanan, Tahap Persebaran Perjalanan, Tahap Pemilihan Moda, dan Tahap Pemilihan Rute.
 - . Pada tahapan bangkitan perjalanan untuk mendapatkan persamaan regresi jumlah perjalanan digunakan analisis persamaan regresi linier.
 - . Variabel-variabel yang dianggap mempengaruhi terbentuknya persamaan regresi jumlah perjalanan tersebut adalah jumlah penduduk, jumlah kendaraan, dan jumlah pendapatan.
- Untuk mendapatkan peramalan data variabel pada nomor tiga digunakan rumus pertumbuhan rata-rata, dimana faktor pertumbuhan tiap zona dalam satu kota dianggap sama
- . Pada tahapan bangkitan perjalanan tidak memperhitungkan perubahan yang terjadi pada tata guna lahan. Tata guna lahan pada tahun rencana dianggap sama dengan tata guna lahan tahun sekarang, dan dianggap kebijakan daerah setempat terhadap tata guna lahan tidak berubah.
 - . Perjalanan yang terjadi merupakan perjalanan berbasis rumah (*Home Base Trip*), dimana semua perjalanan berawal dari rumah dan berakhir di rumah pula.
 - . Jumlah perjalanan pada tahapan bangkitan perjalanan dan distribusi perjalanan merupakan perjalanan yang terjadi tiap hari dalam kondisi normal dan untuk jangka waktu yang panjang.

Perjalanan, Kepemilikan Kendaraan, Jumlah Penduduk, Pendapatan, dan Panjang Perjalanan.

2. Dalam proses pemilihan rute tercepat digunakan algoritma *Dijkstra*.
3. Evaluasi trayek dilihat dari sisi pengguna saja.
4. Dalam proses evaluasi trayek baru hanya didasarkan pada empat faktor, yaitu faktor waktu perjalanan, faktor biaya perjalanan, persentase penggunaan angkutan kota dan pangsa pasarnya.
5. Pembuatan rute baru hanya dilakukan pada wilayah Kota Mojokerto saja, sedangkan wilayah luar Kota Mojokerto tidak diperhitungkan.
6. Dalam pembuatan rute baru dibatasi hanya berdasarkan rute tercepat saja, diasumsikan orang akan cenderung melalui rute tercepat untuk mencapai daerah tujuannya.
7. Dalam pembuatan aplikasi ini dibatasi hanya dapat membuat rute baru berdasarkan rute lama dari tiap trayek angkutan kota, sedangkan untuk pembukaan trayek baru dan penutupan trayek tidak menjadi pembahasan.
8. Pengguna angkutan kota adalah masyarakat umum, dan tidak dibedakan apakah itu pelajar atau bukan.
9. Untuk trayek C, D, E, F dan G dibatasi hanya untuk penggunaan di dalam kota Mojokerto, sedangkan penggunaan di luar Kota Mojokerto tidak diikutsertakan. Sehingga tarif yang diberlakukan dianggap sama dengan trayek A dan B untuk jarak dekat maupun jauh di dalam kota.

22. Dalam pembuatan aplikasi ini tidak memperhitungkan parameter terhadap biaya operasional angkutan kota yang terjadi.
23. Dalam perencanaan trayek angkutan kota ini sepenuhnya menggunakan data dari Dinas Perhubungan, dan proses perolehan data tidak menjadi fokus dalam bahasan ini.
24. Aplikasi ini dibuat untuk melakukan perencanaan satu tahun ke depan.

1. 4. TUJUAN

Tujuan dari tugas akhir ini adalah membuat aplikasi untuk perencanaan trayek angkutan kota di Kota Mojokerto.

1. 5. MANFAAT

Manfaat yang dapat diambil dari pembuatan aplikasi untuk perencanaan trayek angkutan kota ini adalah:

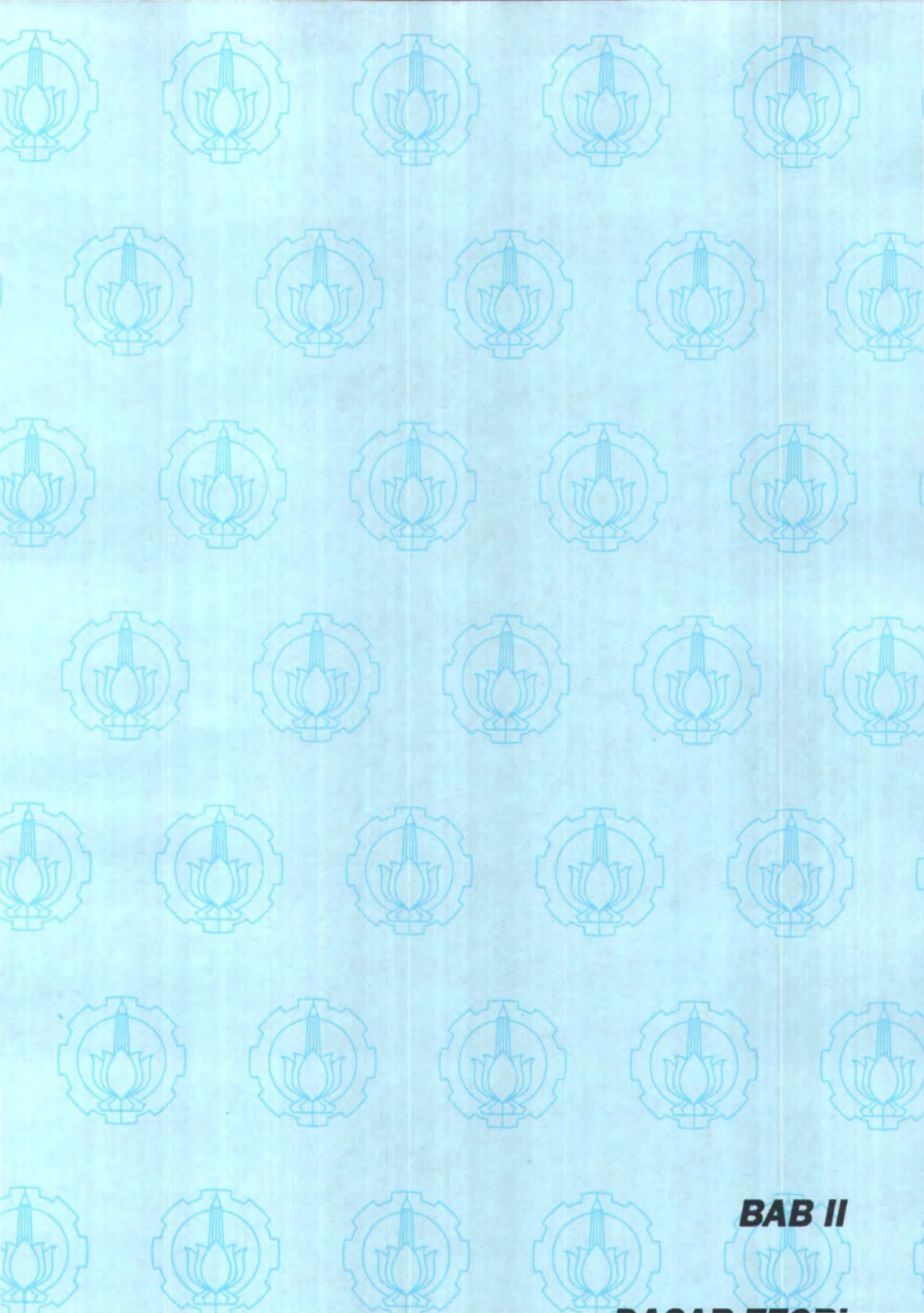
1. Dapat mempermudah dalam melakukan perencanaan trayek angkutan kota berdasarkan pada parameter-parameter yang diberikan.
2. Dapat memberikan hasil evaluasi terhadap hasil perencanaan trayek angkutan kota yang ada, sehingga bisa memberikan penjelasan yang tepat kepada masyarakat tentang adanya perubahan jalur trayek angkutan kota yang telah ada.

1. 6. METODOLOGI PENULISAN TUGAS AKHIR

Metodologi yang digunakan dalam penulisan tugas akhir adalah:

1. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan



BAB II

BAB II

DASAR TEORI

Pada bab ini dijelaskan tentang teori-teori yang berhubungan dengan pembuatan aplikasi untuk perencanaan trayek angkutan kota secara umum. Teori-teori penunjang tersebut merupakan teori yang berhubungan dengan ilmu transportasi pada umumnya dan teori perencanaan trayek angkutan kota pada khususnya. Beberapa teori tersebut adalah peramalan permintaan perjalanan, pola jaringan trayek, dan teori perluasan data sampel. Teori-teori tersebut sebagian besar diambil dari buku Perencanaan dan Pemodelan Transportasi oleh Ofyar Z. Tamin [TAM-2000] dan buku Perencanaan Transportasi untuk Mahasiswa, Perencana, dan Praktisi oleh Fidel Miro [MIR-2005], serta beberapa diktat pengajaran dan buku skripsi dari Sekolah Tinggi Transportasi Darat (STTD) Bekasi, sebagai bahan referensi tambahan. Khusus untuk sub-bab Analisis Regresi-Linier-Berganda beberapa teorinya diambil juga dari buku Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuwan oleh Ronald E Walpole & Raymond H Myers, Bab 10 Regresi Linier Berganda [WAL-1998].

2.1. PERAMALAN PERMINTAAN PERJALANAN

Analisa peramalan permintaan perjalanan, yang merupakan salah satu bagian dalam proses perencanaan trayek angkutan kota, melalui 4 tahapan proses, sehingga sering disebut “Model Perencanaan Transportasi Empat Tahap”, dimana tahapan-tahapan serta urutan terjadinya setiap tahapan tersebut bergantung pada kebutuhan dalam perencanaan trayek angkutan kota. tahapan-tahapan tersebut adalah tahapan pembangkitan perjalanan, tahapan persebaran perjalanan, tahapan pemilihan moda dan tahapan pemilihan rute. Dan pada sub-bab berikut akan

zona lainnya. Tahap pembangkitan perjalanan dalam perencanaan trayek angkutan kota berkaitan dengan karakteristik perjalanan saat ini untuk mengetahui tingkat perjalanan di masa mendatang. Dimana pada tahap pembangkitan perjalanan tersebut diramalkan jumlah perjalanan pada masa mendatang yang dilakukan seseorang pada setiap zona dengan menggunakan rincian data tingkat pembangkitan perjalanan, atribut sosio ekonomi, serta tata guna lahan.

Tahapan pembangkitan perjalanan bertujuan mempelajari dan meramalkan besarnya tingkat pembangkitan perjalanan masa mendatang dengan mempelajari beberapa variasi hubungan antara ciri perjalanan dengan lingkungan tata guna lahan dan atau faktor sosio ekonomi. Dan beberapa kajian transportasi berhasil mengidentifikasi korelasi antara besarnya perjalanan dengan lingkungan tata guna lahan dan faktor sosio ekonomi. Tahapan ini menggunakan data berbasis zona untuk memodel besarnya perjalanan masa mendatang yang akan terjadi, misalnya tata guna lahan, kepemilikan kendaraan, populasi, jumlah pekerja, kepadatan penduduk, pendapatan, dan juga moda transportasi yang digunakan untuk melakukan perjalanan, yang sebelumnya dilakukan analisis dulu terhadap data perjalanan saat ini yang diperoleh dari survey wawancara rumah tangga dan survey wawancara tepi jalan.

2.1.1.1. Analisis Regresi Linier

Terdapat beberapa pemodelan yang dapat digunakan dalam tahapan pembangkitan perjalanan, antara lain pemodelan faktor pertumbuhan dan pemodelan analisis regresi-linier. Tetapi dalam sub bab ini hanya akan dijelaskan pemodelan analisis regresi-linier saja.

sederhana, hubungan secara umum dapat dinyatakan dalam persamaan 2.1 berikut.

$$y = \alpha + \beta x \quad (2.1)$$

Dimana:

y = peubah tidak bebas

x = peubah bebas

α = *intersep* atau konstanta regresi

β = koefisien regresi

Parameter α dan β dapat diperkirakan dengan menggunakan metode kuadrat terkecil yang meminimumkan total kuadratis residual antara hasil model dengan hasil pengamatan. Bila terdapat sampel data $\{(x_i, y_i); i = 1, 2, \dots, n\}$ maka taksiran kuadrat terkecil a dan b dari koefisien regresi α dan β dihitung dengan menggunakan persamaan 2.2 dan 2.3 berikut.

$$b = \frac{n \sum_i (X_i Y_i) - \sum_i (X_i) \cdot \sum_i (Y_i)}{n \sum_i (X_i^2) - \left(\sum_i (X_i) \right)^2} \quad (2.2)$$

$$a = \frac{\sum_{i=1}^n y_i - b \sum_{i=1}^n x_i}{n} \quad (2.3)$$

Persamaan regresi linier sederhana mempunyai suatu taksiran tak bias untuk *variansi* σ^2 yang dapat dilihat pada persamaan 2.4.

y_i = nilai data y ke- i

\hat{Y}_i = nilai taksiran y ke- i

Koefisien determinasi dari persamaan regresi linier sederhana, merupakan nisbah antara simpangan terdefinisi dengan simpangan total, yang dapat dilihat pada persamaan 2.5.

$$R^2 = \frac{\sum_i (\hat{Y}_i - \bar{Y})^2}{\sum_i (Y_i - \bar{Y})^2} \quad (2.5)$$

Dimana:

R^2 = koefisien determinasi

$\sum_i (\hat{Y}_i - \bar{Y})^2$ = simpangan terdefinisi

$\sum_i (Y_i - \bar{Y})^2$ = simpangan total

\hat{Y} = nilai Y taksiran

\bar{Y} = nilai Y rata-rata



2. 1. 1. 2. Analisis Regresi Linier Berganda

Pengembangan lanjut dari regresi-linier, seperti yang telah dijelaskan pada sub-bab sebelumnya adalah regresi-linier-berganda, yang digunakan untuk kasus dengan peubah bebas yang lebih banyak. Bentuk umum dari metode analisis regresi linear berganda dapat dinyatakan pada persamaan 2.6 sebagai berikut.

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k \quad (2.6)$$

$\beta_1 \dots \beta_k$ = koefisien regresi

Terdapat beberapa metode yang dapat digunakan untuk menyelesaikan persamaan regresi linier berganda, yaitu dengan menggunakan pendekatan kuadrat terkecil dan dengan menggunakan perhitungan matrik. Dalam sub-bab ini hanya akan dibahas penyelesaian persamaan regresi linier berganda dengan menggunakan perhitungan matrik saja.

Dalam menyelesaikan persamaan regresi linier berganda, khususnya dalam hal ini untuk peubah lebih dari dua, maka teori matrik dapat menyederhanakan perhitungan. Misalnya terdapat k peubah bebas, x_1, x_2, \dots, x_k dan n pengamatan y_1, y_2, \dots, y_n masing-masing dapat dinyatakan dengan persamaan 2.7.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \varepsilon_i \quad (2.7)$$

Model ini pada dasarnya menyatakan n persamaan yang memberikan bagaimana nilai respon diperoleh dalam proses penelitian. Dengan menggunakan lambang matriks, persamaan kuadrat dapat ditulis seperti pada persamaan 2.8.

$$y = X\beta + \varepsilon \quad (2.8)$$

Dimana:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad x = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}$$

Parameter β_0 sampai dengan β_k dapat dicari dengan menggunakan perhitungan matrik, seperti dapat dilihat pada persamaan 2.9.

$$A = X'X = \begin{bmatrix} n & \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{2i} & \cdots & \sum_{i=1}^n x_{ki} \\ \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{1i}^2 & \sum_{i=1}^n x_{1i}x_{2i} & \cdots & \sum_{i=1}^n x_{1i}x_{ki} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_{ki} & \sum_{i=1}^n x_{ki}x_{1i} & \sum_{i=1}^n x_{ki}x_{2i} & \cdots & \sum_{i=1}^n x_{ki}^2 \end{bmatrix} \quad (2.10)$$

$$g = X'y = \begin{bmatrix} g_0 = \sum_{i=1}^n y_i \\ g_1 = \sum_{i=1}^n x_{1i}y_i \\ \vdots \\ g_2 = \sum_{i=1}^n x_{ki}y_i \end{bmatrix} \quad (2.11)$$

Maka persamaan normal dapat dinyatakan dalam bentuk matriks seperti persamaan 2.12

$$A\beta = g \quad (2.12)$$

Bila matriks A tak singular, maka koefisien regresi β dapat dicari seperti pada persamaan 2.13.

$$\beta = A^{-1}g = (X'X)^{-1}X'y \quad (2.13)$$

Analisis regresi-linear-berganda adalah suatu metode statistik. Untuk menggunakannya terdapat beberapa asumsi yang perlu diperhatikan:

- Nilai peubah, khususnya peubah bebas, mempunyai nilai tertentu atau merupakan nilai yang didapat dari hasil survey tanpa kesalahan berarti,
- Peubah tidak bebas (y) harus mempunyai hubungan korelasi linear dengan peubah bebas (x). jika hubungan tersebut tidak linear, transformasi linear harus dilakukan, meskipun batasan ini mempunyai implikasi lain dalam

- Nilai peubah tidak bebas harus tersebar normal atau minimal mendekati normal,
- Nilai peubah bebas sebaiknya merupakan besaran yang relatif mudah diproyeksikan.

Pada analisis regresi-linier-berganda mempunyai solusi yang tetap sama dengan analisis regresi-linier, hanya saja analisis regresi-linier-berganda lebih kompleks sehingga beberapa hal baru harus dipertimbangkan sebagai berikut:

- Multikolinear*, hal ini terjadi karena adanya hubungan linier antar-peubah. Pada kasus ini, beberapa persamaan yang mengandung b tidak saling bebas dan tidak dapat dipecahkan secara unik.
- Jumlah parameter ' b ' yang dibutuhkan, untuk memutuskan hal ini, beberapa faktor harus dipertimbangkan.
 - Apakah ada alasan teori yang kuat sehingga harus melibatkan peubah itu atau apakah peubah itu penting untuk proses uji dengan model tersebut.
 - Apakah peubah itu signifikan dan apakah tanda koefisien parameter yang didapat sesuai dengan teori atau intuisi.

Jika diragukan, terapkan salah satu cara, yaitu menghilangkan peubah itu dan melakukan proses regresi lagi untuk melihat efek dibuangnya peubah itu terhadap peubah lainnya yang masih digunakan oleh model tersebut. Jika ternyata tidak terlalu terpengaruh, peubah itu dibuang saja sehingga kita mendapatkan model yang lebih sederhana dan dapat ditaksir secara lebih tepat. Terdapat pendekatan yang dapat digunakan, yaitu pendekatan langkah-demi-langkah atau *stepwise*.

iii. $r = -1$, menunjukkan adanya hubungan yang sangat kuat dan berbanding terbalik antara variabel-variabelnya atau dikatakan berkorelasi negatif sempurna.

Taksiran tak bias untuk variansi σ^2 dari persamaan regresi linier berganda dapat dilihat pada persamaan 2.16.

$$s^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - k - 1} \quad (2.16)$$

Dimana:

s^2 = taksiran tak bias untuk variansi σ^2

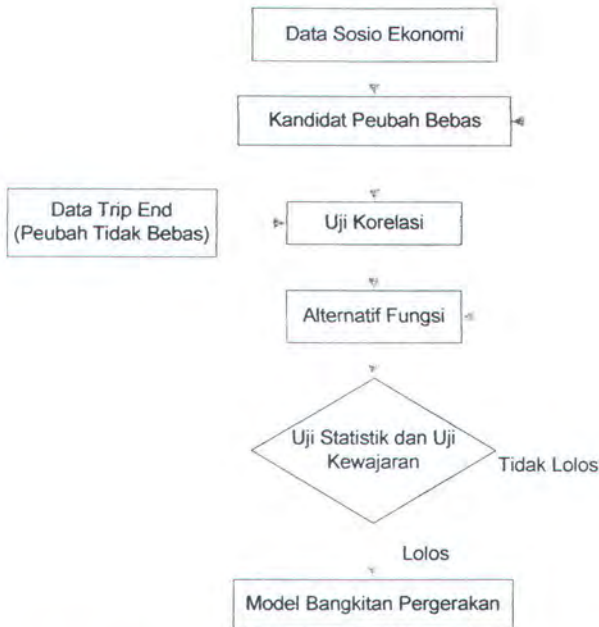
n = banyaknya data

k = banyaknya peubah bebas

y_i = nilai data y ke- i

\hat{y}_i = nilai taksiran y ke- i

Uji t-test, dapat digunakan untuk dua tujuan, yaitu untuk menguji signifikan nilai koefisien korelasi (r) dan untuk menguji signifikan nilai koefisien regresi. Setiap peubah yang mempunyai koefisien regresi yang tidak signifikan secara statistik harus dibuang dari model.



Gambar 2.1 Proses Kalibrasi dan Pengabsahan Model Analisis Regresi

(Sumber: O.Z. Tamin, *Perencanaan Dan Pemodelan Transportasi*)

2. 1. 1. 3. Uji Statistika

Dalam membuat suatu model analisis-regresi terdapat beberapa uji statistik yang harus dilakukan agar model regresi yang dihasilkan dapat dinyatakan absah. Beberapa uji statistik tersebut adalah:

- Uji Korelasi, digunakan untuk menguji signifikan nilai koefisien korelasi. Uji korelasi dinyatakan dalam bentuk persamaan 2.17.

$$t_o = \frac{r\sqrt{n-2}}{\sqrt{(1-r^2)}} \quad (2.17)$$

Dimana:

- i Menyatakan hipotesis nol dan hipotesis alternatif
 H_0 : korelasi = 0
 H_1 : ada korelasi
 - ii Menentukan tingkat signifikan dengan derajat keyakinan $n-2$ dan $\alpha = 5\%$
 - iii Menentukan kriteria uji
 H_0 : diterima bila $-t_{\alpha/2} \leq t_o \leq t_{\alpha/2}$
 H_1 : diterima bila $t_o < -t_{\alpha/2}$ atau $t_o > t_{\alpha/2}$
 - iv Menarik kesimpulan
- b. Uji t-test untuk regresi linier sederhana, digunakan untuk menguji signifikan nilai koefisien regresi dan konstanta regresi pada persamaan regresi linier sederhana. Setiap peubah yang mempunyai koefisien regresi dan konstanta regresi yang tidak signifikan secara statistik harus dibuang dari model. Persamaan uji-t untuk menguji nilai koefisien regresi dapat dilihat pada 2.18

$$t = \frac{b_0}{s / \sqrt{J_{xx}}} \quad (2.18)$$

Dimana:

$s = \sqrt{s^2}$, s^2 dapat dicari dengan menggunakan persamaan 2.4

b_0 = koefisien regresi

J_{xx} , dapat dicari dengan menggunakan persamaan 2.19 seperti berikut.

$$J_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i \right)^2}{n} \quad (2.19)$$

- ii Menentukan tingkat signifikan dengan derajat keyakinan $(n-2)$ dan $\alpha = 5\%$
- iii Menentukan kriteria uji

H_0 : diterima bila $-t_{\alpha/2} \leq t \leq t_{\alpha/2}$

H_1 : diterima bila $t < -t_{\alpha/2}$ atau $t > t_{\alpha/2}$

- iv Menarik kesimpulan

Sedangkan persamaan uji-t untuk menguji nilai konstanta regresi dapat dilihat pada persamaan 2.20.

$$t = \frac{a}{s \sqrt{\sum_{i=1}^n x_i^2 / nJ_{xx}}} \quad (2.20)$$

Dengan prosedur hipotesisnya adalah sebagai berikut:

- i Menyatakan hipotesis nol dan hipotesis alternatif
- $H_0: \alpha = 0$, konstanta regresi tidak signifikan
- $H_1: \alpha \neq 0$, konstanta regresi signifikan
- ii Menentukan tingkat signifikan dengan derajat keyakinan $(n-2)$ dan $\alpha = 5\%$
 - iii Menentukan kriteria uji

H_0 : diterima bila $-t_{\alpha/2} \leq t \leq t_{\alpha/2}$

H_1 : diterima bila $t < -t_{\alpha/2}$ atau $t > t_{\alpha/2}$

- iv Menarik kesimpulan

c. Uji t-test, digunakan untuk menguji signifikan nilai koefisien regresi yang

Ho : tidak ada hubungan antara variabel

H1 : ada hubungan

- ii Menentukan tingkat signifikan dengan derajat keyakinan (k , $n-k-1$) atau $\alpha = 5\%$
- iii Menentukan kriteria uji
 - Ho : diterima bila $F_{\text{tabel}} > F_{\text{hitung}}$
 - H1 : diterima bila $F_{\text{hitung}} > F_{\text{tabel}}$
- iv Menarik kesimpulan

2. 1. 1. 4. Metode Analisis Langkah Demi Langkah Tipe 1

Dalam melakukan analisis bangkitan pergerakan dengan menggunakan model analisis-regresi berbasis zona, terdapat tiga metode analisis yang dapat digunakan, yaitu:

- Metode analisis langkah demi langkah tipe 1
- Metode analisis langkah demi langkah tipe 2
- Metode coba-coba

Pada sub bab ini hanya akan dijelaskan metode analisis langkah demi langkah tipe 1. Dimana metode ini secara bertahap mengurangi jumlah peubah bebas sehingga didapatkan model terbaik yang hanya terdiri dari beberapa peubah bebas. Berikut tahapan-tahapan dari metode tersebut.

- a. Tahap 1. Tentukan parameter sosio-ekonomi yang akan digunakan sebagai

- Variabel bebas (X) yang diuji tersebut harus mempunyai hubungan kedekatan (korelasi) dengan variabel tidak bebasnya (Y), variabel bebas yang tidak mempunyai korelasi dengan variabel tidak bebasnya tidak dapat digunakan dalam persamaan regresi,
 - Antara variabel bebas (X) yang diuji tidak boleh terdapat hubungan korelasi karena hal ini untuk menghindari terjadinya saling membiaskan (*distorsi*) antara variabel-variabel tersebut,
 - Jika terjadi *distorsi* antara variabel bebas yang diuji, maka harus dipilih salah satu variabel bebas (X) tersebut. Dimana pemilihan ini diambil dari koefisien korelasi terbesar antara variabel bebas (X) terhadap variabel tidak bebas (Y)
- b. Tahap 2. Lakukan analisis regresi-linier-berganda dengan semua peubah bebas terpilih untuk mendapatkan nilai koefisien *determinasi* serta nilai nilai konstanta dan koefisien regresinya.
- c. Tahap 3. Tentukan parameter yang mempunyai korelasi terkecil terhadap peubah tidak bebasnya dan hilangkan parameter tersebut. Lakukan kembali analisis regresi-linier-berganda dan dapatkan kembali nilai koefisien *determinasi* serta konstanta dan koefisien regresinya.
- d. Tahap 4. Lakukan kembali tahap 3 satu demi satu sampai hanya tertinggal satu parameter saja.
- e. Tahap 5. Kaji nilai koefisien *determinasi* serta nilai konstanta dan koefisien regresi setiap tahap untuk menentukan model terbaik dengan kriteria berikut:
- Semakin banyak peubah bebas yang digunakan, semakin baik model

adalah matriks berdimensi dua yang berisi informasi mengenai besarnya pergerakan antarlokasi (zona) di dalam daerah tertentu. Baris menyatakan zona asal dan kolom menyatakan zona tujuan, sehingga sel matriks-nya menyatakan besarnya arus dari zona asal ke zona tujuan. Dalam hal ini notasi T_{id} menyatakan besarnya arus pergerakan yang bergerak dari zona asal i ke zona tujuan d selama selang waktu tertentu.

Pola pergerakan dapat dihasilkan jika suatu MAT dibebankan ke suatu sistem jaringan transportasi. Dengan mempelajari pola pergerakan yang terjadi, seseorang dapat mengidentifikasi permasalahan yang timbul sehingga beberapa solusi segera dapat dihasilkan. MAT dapat memberikan indikasi rinci mengenai kebutuhan akan pergerakan sehingga MAT memegang peran yang sangat penting dalam berbagai kajian perencanaan dan manajemen transportasi.

Metode untuk mendapatkan MAT dapat dikelompokkan menjadi dua bagian utama, yaitu metode konvensional dan metode tidak-konvensional. Pengelompokkan tersebut dapat dilihat pada gambar 2.2. Pada sub-bab ini hanya akan dijelaskan secara detail Metode Analogi saja.

Metode analogi merupakan suatu metode yang berasumsi bahwa pola pergerakan pada saat sekarang dapat diproyeksikan ke masa mendatang dengan menggunakan tingkat pertumbuhan zona yang berbeda-beda. Semua metodenya mempunyai persamaan umum seperti pada persamaan 2.24.

$$T_{id} = t_{id} \cdot E \quad (2.24)$$

Dimana:

T_{id} = pergerakan pada masa mendatang dari zona asal i ke zona tujuan d

untuk semua daerah kajian atau untuk zona tertentu saja yang kemudian digunakan untuk mendapatkan MAT.

Metode analogi dapat dikelompokkan menjadi tiga kelompok utama, yaitu:

Metode Tanpa-Batasan

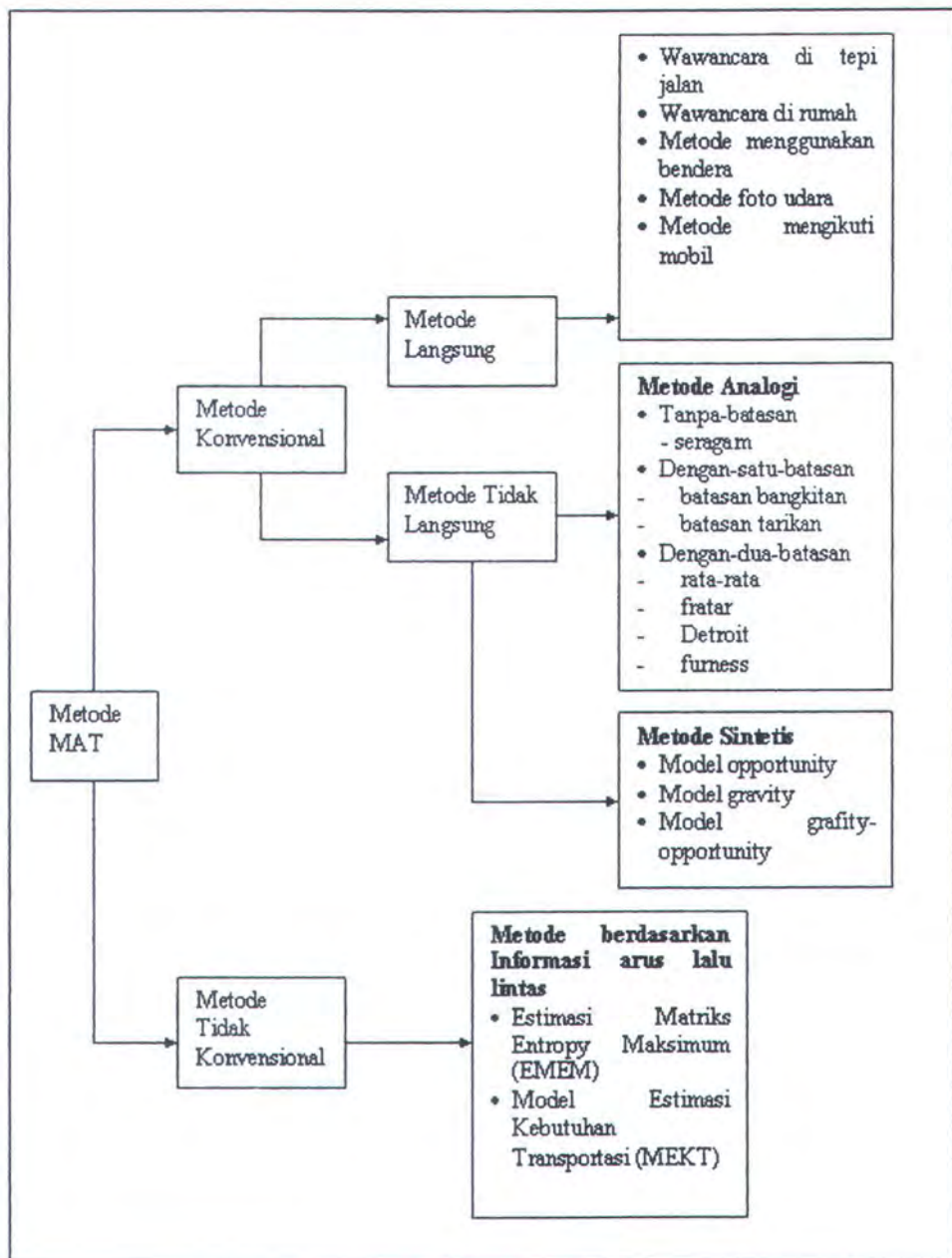
Metode ini adalah metode tertua dan paling sederhana. Dalam metode ini diasumsikan bahwa untuk keseluruhan daerah kajian hanya ada satu nilai tingkat pertumbuhan yang digunakan untuk mengalikan semua pergerakan pada saat sekarang untuk mendapatkan pergerakan pada masa mendatang. Metode ini tidak menjamin bahwa total pergerakan yang dibangkitkan dari setiap zona asal dan total pergerakan yang tertarik ke setiap zona tujuan akan sama dengan total bangkitan dan tarikan yang diharapkan pada masa mendatang.

Metode Dengan-Satu-Batasan

Metode dengan satu batasan mempunyai dua jenis metode untuk batasannya, yaitu metode dengan-batasan-bangkitan dan metode dengan-batasan-tarikan.

- Metode Dengan-Batasan-Bangkitan

Metode ini digunakan jika informasi yang tersedia adalah perkiraan bangkitan pergerakan pada masa mendatang, sedangkan perkiraan tarikan pergerakan tidak tersedia atau dapat juga tersedia tetapi dengan tingkat akurasi yang rendah.



- Untuk setiap zona, jumlah hasil pendekatan pertama dibagi dengan total pergerakan yang diperkirakan (dihasilkan dari tahapan bangkitan pergerakan), untuk mendapatkan nilai tingkat pertumbuhan yang baru yang selanjutnya digunakan sebagai pendekatan kedua.
- Pergerakan yang dihasilkan pada pendekatan pertama yang kemudian disebarakan, dan ini sebanding dengan pergerakan pada masa sekarang dan nilai tingkat pertumbuhan yang baru (hasil pendekatan pertama). Kedua nilai ini kemudian dirata-ratakan dan proses diulangi sampai tercapai kesesuaian antara pergerakan yang dihitung dengan yang diinginkan.

- Metode *Detroit*

Metode ini dikembangkan bersamaan dengan pelaksanaan pekerjaan *Detroit Metropolitan Area Traffic Study* dalam usaha mengatasi kekurangan metode sebelumnya dan sekaligus mengurangi waktu operasi komputer. Prosesnya mirip dengan metode rata-rata dan *Fratar*, tetapi mempunyai asumsi walaupun jumlah pergerakan dari zona i meningkat sesuai dengan tingkat pertumbuhan E_i , pergerakan ini harus juga disebarakan ke zona d sebanding dengan E_d dibagi dengan tingkat pertumbuhan global (E)

- Metode *Furness*

Furness (1965) mengembangkan metode yang pada saat sekarang sangat sering digunakan dalam perencanaan transportasi. Metodanya sangat sederhana dan mudah digunakan. Pada metode ini

Terdapat beberapa keuntungan dan kerugian penggunaan metode Analogi, yang menjadikan alasan metode ini digunakan atau tidak digunakan dalam perencanaan transportasi. Beberapa keuntungan metode analogi adalah sebagai berikut:

- Mudah dimengerti dan digunakan, hanya membutuhkan data pergerakan antarzona (MAT) pada masa sekarang dan perkiraan tingkat pertumbuhan zona pada masa mendatang yang sederhana;
- Proses pengulangannya sederhana;
- Data aksesibilitas (waktu, jarak, dan biaya) antarzona tidak diperlukan;
- Penggunaannya fleksibel, misalnya untuk moda transportasi lain, untuk tujuan perjalanan yang berbeda, untuk selang waktu yang berbeda, dan juga dapat digunakan untuk arah pergerakan yang berbeda;
- Sudah sering diabsahkan dan menghasilkan tingkat ketepatan yang cukup tinggi jika digunakan pada daerah yang pola pengembangan wilayahnya stabil.

Akan tetapi, selain keuntungan, terdapat juga beberapa permasalahan yang sering timbul dalam pemakaiannya. Di antaranya adalah yang berikut ini:

- Metode ini membutuhkan masukan data lengkap dari seluruh pergerakan antarzona pada saat sekarang (t_{id}); informasi ini tentu sangat mahal.
- Dibutuhkan jumlah zona yang selalu tetap; dengan kata lain, tidak boleh ditambah dengan zona baru.
- Kelemahan lainnya adalah jika ditemukan bahwa antara dua buah zona pada saat sekarang belum terjadi pergerakan ($t_{id} = 0$) atau mungkin karena ada galat

Kelemahan lain, jika pada masa sekarang terdapat sel matriks yang tidak didapatkan informasi pergerakannya (datanya tidak ada), maka sel matriks tersebut tidak akan pernah bisa didapatkan pergerakan masa mendatangnya. Karena itu, metode ini tidak dapat digunakan untuk melengkapi sel matriks yang kosong dengan menambahkannya dari matriks parsial.

Metode ini tidak bisa digunakan untuk daerah yang pada masa mendatang mengalami perubahan aksesibilitas yang nyata pada sistem jaringan transportasinya, misalnya pelebaran jalan, pembangunan jalan baru, dan pembangunan jalan bebas hambatan. Jadi, model ini tidak cocok untuk peramalan untuk waktu yang cukup panjang.

Untuk selang waktu yang pendek dan di daerah yang stabil pengembangan wilayahnya, metode ini dapat digunakan dengan baik. Sebaliknya, metode ini tidak dapat digunakan pada daerah yang pesat pengembangan wilayahnya dan tajam peningkatan aksesibilitas sistem jaringan transportasinya.

1. 3. Tahapan Pemilihan Moda

Sebelum masuk ke dalam proses analisis pilihan moda, harus dilakukan mengelompokkan beberapa hal ke dalam beberapa kelompok, yaitu:

1. Pengguna Jasa Transportasi/Pelaku Perjalanan (*TRIP MAKER*)

Masyarakat pelaku perjalanan dapat dikelompokkan menjadi dua kelompok, yaitu:

- Penumpang tergantung pada angkutan umum (*Captive Transit Raider*),

pula menggunakan angkutan umum. Dimana kelompok ini merupakan jumlah terbesar pada negara maju.

Bentuk Alat Transportasi/Jenis Pelayanan Transportasi

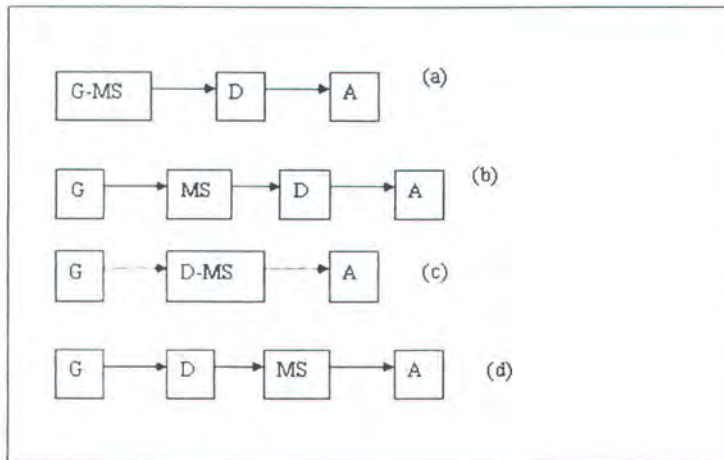
Secara umum ada dua kelompok besar moda transportasi, yaitu:

- Kendaraan pribadi (*Private Transportation*), yaitu moda transportasi yang dikhususkan buat pribadi seseorang, dan seseorang itu bebas memakainya kemana saja.
- Kendaraan Umum (*Public Transportation*), yaitu moda transportasi yang diperuntukkan buat bersama (orang banyak), kepentingan bersama, menerima pelayanan bersama, mempunyai arah dan titik tujuan yang sama, serta terikat dengan peraturan trayek yang sudah ditentukan dan jadwal yang sudah ditetapkan dan para pelaku perjalanan wajib menyesuaikan diri dengan ketentuan tersebut apabila angkutan umum ini sudah mereka pilih.

Tahap pemilihan moda atau moda split ini mempunyai 4 jenis model yang berkaitan dengan tahap-tahap perencanaan transportasi, yaitu:

Model Jenis I (*Dual Generation Gravity Model*), dimana tahapan moda split digabungkan dengan tahapan trip generation atau bangkitan perjalanan, untuk lebih jelasnya dapat dilihat pada gambar 2.3 (a)

Model Jenis II (*Trip End Model*), dimana tahapan moda split dilakukan sebelum tahapan distribusi perjalanan, untuk lebih jelasnya dapat dilihat pada



Gambar 2.3 Model Tahapan Perencanaan Transportasi

eterangan Gambar 2.3:

- G = Bangkitan Perjalanan (*Generation*)
- D = Penyebaran Perjalanan (*Distribution*)
- MS = Pemilihan Moda (*Moda Split*)
- A = Pemilihan Rute (*Assignment*)

Untuk mendapatkan hasil perilaku jumlah pelaku perjalanan yang menggunakan dua atau lebih moda transportasi yang betul-betul proporsional, lakukan beberapa tahapan analisis^[1], yaitu:

Tahapan Pertama, pengidentifikasian beberapa faktor (variabel) yang diasumsikan berpengaruh secara berarti terhadap perilaku pelaku perjalanan (*trip maker behavior*) dalam menjustifikasi pilihan alternatif alat angkutan yang dipakai untuk berpergian

model pilihan moda angkutan seperti “*binary modal*” diantaranya *logit biner*, *probit*, *multinomial logit*, atau *Gunarson* dengan cara mengeksponenkan nilai kepuasan masing-masing moda angkutan yang sudah kita dapatkan pada tahapan kedua. Atau dapat juga digunakan Model Ujung Perjalanan (*Trip End Model*), yang dilakukan dengan memodelkan menggunakan analisis regresi linier berganda.

Tahapan Keempat, merupakan tahapan terakhir, pada tahapan ini barulah didapati angka proporsi (dalam %) peluang atau pangsa pasar masing-masing moda angkutan untuk dipilih dari sejumlah calon pengguna moda (*user*) tertentu sebagai perkiraan (*estimation*) serta angka mutlakny.

Dalam bahasan sub-bab ini analisis tahapan kedua tidak digunakan, karena pada tahapan ketiga untuk mendapatkan model persamaan peluang alternatif pilihan moda angkutan tidak menggunakan *Binary Modal*, melainkan menggunakan model ujung perjalanan, sehingga tidak memerlukan nilai kepuasan (*utility*) yang terdapat pada tahapan kedua untuk memperoleh model peluang.

1.3.1. Tahapan I

Identifikasikan berbagai faktor dan variabel yang diasumsikan berpengaruh terhadap perilaku pelaku perjalanan. Ada empat kelompok faktor yang dianggap kuat pengaruhnya terhadap perilaku pelaku perjalanan, yaitu:

1. Kelompok Faktor Karakteristik Perjalanan, terdapat beberapa variabel yang dianggap kuat pengaruhnya terhadap perilaku pengguna jasa moda transportasi dalam memilih moda angkutan, yaitu:

- Variabel Pendapatan
- Variabel Kepemilikan Kendaraan
- Variabel Kondisi Kendaraan Pribadi
- Variabel Kepadatan Pemukiman
- Variabel Sosial Ekonomi, seperti struktur dan ukuran keluarga, jenis kelamin, usia, jenis pekerjaan, lokasi pekerjaan, kepemilikan terhadap lisensi mengemudi (SIM).

4. Kelompok Faktor Karakteristik Sistem Transportasi, pada kelompok faktor ini, seluruh variabel yang berpengaruh terhadap perilaku si pembuat perjalanan dalam memilih moda transportasi berhubungan dengan kinerja pelayanan sistem transportasi seperti berikut:

- Variabel Waktu Perjalanan (*Relative Travel Time*) mulai dari lamanya menunggu kendaraan di terminal, waktu jalan ke terminal, dan waktu di atas kendaraan. Variabel waktu ini kemudian dibandingkan dengan waktu perjalanan menggunakan alternatif moda lain, sehingga timbul suatu kompetisi sehat antara moda yang satu dengan alternatif moda yang lain, untuk saling menawarkan pelayanan terbaik (waktu yang tersingkat) kepada calon pelaku perjalanan. *National Capital Transportation Agency* (NCTA) mengembangkan suatu model yang disebut dengan Rasio Waktu Perjalanan (RWP) yang hanya membandingkan gambaran waktu perjalanan antara moda transportasi umum dengan moda transportasi

$$RPP = \frac{x_2 + x_3 + x_4 + x_5}{x_7 + x_8} \quad (2.28)$$

Dimana:

RPP = Rasio pelayanan Perjalanan

Rasio-rasio yang dimodelkan oleh NCTA tersebut selanjutnya, dapat dijadikan evaluasi bagi kinerja trayek atau moda angkutan umum terhadap moda angkutan pribadi. Dengan nilai semakin kecil atau kurang dari satu berarti bahwa kinerja moda angkutan umum semakin baik dari kinerja moda angkutan pribadi.

4. Kelompok Faktor Karakteristik Kota dan Zona, variabel yang ada dalam kelompok ini antara lain adalah:

- Variabel Jarak Kediaman dengan tempat kegiatan (CBD)
- Variabel Kepadatan Penduduk.

4. 1. 3. 2. Tahapan III

Pada tahapan ketiga dilakukan pemodelan peluang masing-masing alternatif pilihan moda yang akan dipakai melalui beberapa bentuk model pendekatan. Terdapat beberapa model yang dapat digunakan, yaitu salah satunya adalah model ujung perjalanan (*Trip End Model*). Proses perhitungan menggunakan model ini sederhana, hanya menghitung persentase perjalanan dari

1. 4. Tahapan Pemilihan Rute

Terdapat sebuah algoritma dasar yang sering digunakan untuk mencari rute tercepat (atau termurah) dalam suatu jaringan jalan. Algoritma itu adalah algoritma *Dijkstra* (1959)^[2], yang diterangkan dengan menggunakan notasi perorientasi simpul: jarak (biaya) ruas antara dua titik **A** dan **B** dalam suatu jaringan dinotasikan dengan $d_{A,B}$. Rute didefinisikan dalam bentuk urutan simpul yang saling berhubungan, **A-C-D-H** dan seterusnya, sedangkan jarak rute adalah penjumlahan setiap ruas yang ada dalam rute tersebut. Selanjutnya akan dijelaskan lebih detail mengenai algoritma *Dijkstra*^[15] untuk mencari rute tercepat antara dua buah pusat zona.

Algoritma *Dijkstra* adalah suatu algoritma yang berfungsi untuk mencari rute dengan bobot paling minimum antara dua buah titik simpul atau dua buah *node*. Cara kerja algoritma ini adalah dengan cara menandai setiap simpul yang terpilih dengan permanen label sampai semua simpul mendapat permanen label. Kemudian semua lintasan dibandingkan dan dicari lintasan yang memiliki total bobot minimum.

Algoritma ini pada tiap iterasinya memeriksa sisi-sisi yang menghubungkan simpul-simpul pada W dan simpul-simpul pada $(V-W)$, dan memindahkan simpul-simpul w dari $(V-W)$ ke W yang memenuhi kriteria tertentu, yaitu w dengan bobot yang paling minimum yang menghubungkan ke suatu simpul tertentu di $(V-W)$ sehingga bobot dari simpul asal V_s ke simpul tersebut adalah minimal. Dalam pengaplikasiannya penghitungan bobot dari simpul asal V_s dapat disederhanakan dengan memberikan suatu nilai yang disebut

simpul pada (V-W) masih perlu diganti pada setiap iterasinya untuk mendapatkan w seperti diterangkan di atas. Setiap mendapatkan w maka nilai *minpath* diubah dengan nilai yang paling minimum antara nilai *minpath* lama pada simpul tersebut dengan nilai *minpath* baru yang diperoleh dari penjumlahan nilai *minpath* pada simpul sebelumnya dijumlahkan dengan panjang dari simpul sebelumnya dengan simpul tersebut. Dimana sebelumnya telah diinisialisasikan terlebih dahulu bahwa nilai *minpath* pada semua simpul adalah positif tak berhingga. Selengkapnya algoritma *Dijkstra* adalah sebagai berikut.

Inisialisasi awal sebagai berikut:

- i. Simpul-simpul dalam W mula-mula hanya berisi Vs yang merupakan simpul asal.
- ii. Kemudian isi *minpath* tiap simpul V dengan
 - o Jika terdapat sisi yang menghubungkan antara simpul Vs dengan simpul V isi *minpath* dengan beban yang dimiliki, $Weight[V_s, V]$
 - o Dan isi dengan $+\infty$, jika sisi Vs ke V tidak ada.

Kemudian iterasi mulai dilakukan hingga (V-W) tak tersisa atau V_e (simpul tujuan) ditemukan, dengan langkah-langkah sebagai berikut:

- i. Dari nilai *minpath* tiap simpul cari simpul w dalam (V-W), yaitu simpul yang memiliki nilai *minpath* terkecil dan bukan tak hingga.
- ii. Jika simpul w tersebut ada maka:

iv. Nilai W akhir adalah rute dengan bobot paling minimum.

2. 2. PERLUASAN DATA

Data sampel yang diperoleh dari hasil survey sebelum dianalisa lebih lanjut terlebih dahulu harus dilakukan ekspansi atau perluasan data sampel menjadi data populasi. Terdapat dua jenis survey yang harus diekspansi untuk mendapatkan data populasinya, yaitu survey wawancara rumah tangga (*Home Interview / HI*) dan survey wawancara tepi jalan (*Road Side Interview / RSI*). Dan pada sub-bab berikut akan dijelaskan perluasan data sampel untuk masing-masing survey yang dilakukan.

Data sampel hasil survey HI sebelum dilakukan analisa, harus dilakukan ekspansi atau perluasan data sampel menjadi data populasi terlebih dahulu. Nilai ekspansi ini dihitung untuk tiap zona yang disurvey. Dimana rumus untuk perluasan data sampel survey HI menjadi data populasi survey HI, yaitu besarnya perjalanan yang dilakukan orang dalam satu hari di suatu wilayah zona disurvey dari satu zona ke zona lainnya, dapat dilihat pada persamaan 2.31.

$$\text{Data Populasi survey HI} = \text{Data Sampel survey HI} * E \quad (2.31)$$

Dimana:

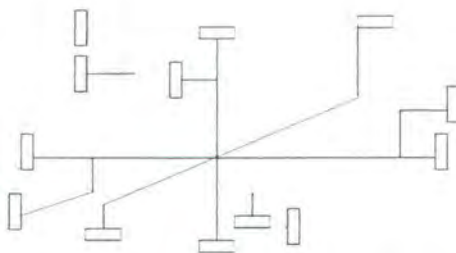
E = faktor ekspansi, yang diperoleh dari persamaan 2.32.

$$E = \frac{A - A / B.(C + D)}{B - C - D} \quad (2.32)$$

Dimana:

Pola *Radial*

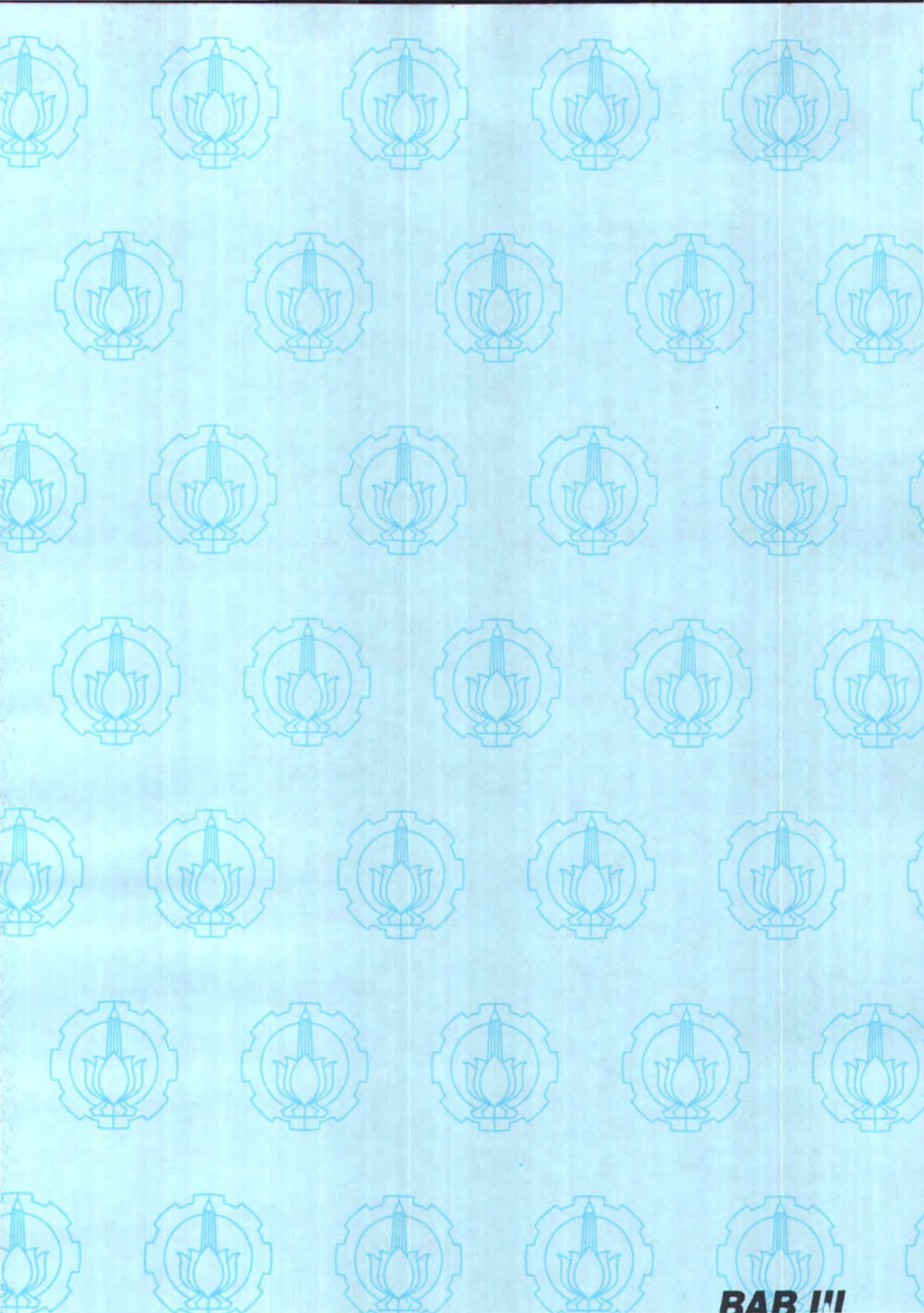
Pada pola *radial*, seluruh atau hampir seluruh jalur utama membentuk jari-jari dari pusat kota ke daerah pinggir kota. Pelayanan trayek memotong pusat kota, memutar pusat kota atau berhenti di pusat kota. Dapat dilihat pada gambar 2.4 bahwa jalur utama selalu saling memotong pada satu titik pusat yang merupakan pusat kota, dan dari titik pusat tersebut terdapat garis-garis membentuk seperti jari-jari yang merupakan jalur utama yang menghubungkan antara pusat kota dengan daerah pinggiran kota. Keuntungan dari sistem ini jumlah titik perpindahan sedikit karena mayoritas penumpang menuju satu titik, sedangkan kerugiannya adalah menambah kemacetan pada daerah pusat kota.



Gambar 2.4 Jaringan Trayek Pola *Radial*

Pola *Orthogonal/Grid*

Pola *Orthogonal/Grid* ditandai dengan lintasan-lintasan yang membentuk *grid* (kisi-kisi) sebagian menuju pusat kota dan sebagian lainnya tidak melalui pusat kota. Tujuan utama pola ini memberikan pelayanan yang sama untuk semua bagian kota. Pola ini dapat dilihat pada gambar 2.5, dimana



BAB III

ANALISIS KEBUTUHAN DAN PERANCANGAN APLIKASI

Pada bab ini dijelaskan tentang analisis kebutuhan sistem dan perancangan aplikasi. Analisis kebutuhan sistem merupakan uraian tentang hal-hal yang diperlukan oleh sistem. Setelah itu dilanjutkan dengan perancangan aplikasi sistem yang disesuaikan dengan daftar kebutuhan sistem pada analisis kebutuhan sistem yang telah disusun sebelumnya. Perancangan aplikasi untuk sistem tersebut terdiri dari perancangan data, proses dan antarmuka. Perancangan data menggunakan *entity relational modelling* yang menghasilkan *Entity Relational Database* (ERD), dilanjutkan dengan melakukan pemetaan ke dalam bentuk tabel, dan tahap terakhir adalah melakukan normalisasi terhadap tabel-tabel hasil pemetaan. Setelah perancangan data dilanjutkan dengan perancangan proses yang dilakukan secara terstruktur, dari proses secara umum sampai dengan proses detailnya, dan perancangan tersebut disajikan dengan spesifikasi proses dalam bentuk *Data Flow Diagram* (DFD). Perancangan yang terakhir dilakukan adalah perancangan antar muka dengan menampilkan *layout-layout* yang nantinya akan dijadikan antar muka antara aplikasi dengan pengguna.

3.1. ANALISIS KEBUTUHAN SISTEM

Analisis kebutuhan sistem membahas secara garis besar kondisi sistem yang ada saat ini, kondisi sistem baru yang diinginkan, dan uraian kebutuhan-kebutuhan sistem atau *user requirement* untuk membangun sebuah sistem baru seperti yang diinginkan. Pembahasan-pembahasan tersebut dimulai dari pembahasan tentang gambaran kondisi yang sedang berjalan saat ini. Kemudian

- b Tahun 1996 jumlahnya sebesar 105.421 jiwa
- c Tahun 1997 jumlahnya sebesar 105.981 jiwa
- d Tahun 1998 jumlahnya sebesar 107.123 jiwa
- e Tahun 1999 jumlahnya sebesar 107.925 jiwa
- f Tahun 2000 jumlahnya sebesar 108.346 jiwa

Tabel 3.1 Daftar Kecamatan dan Desa di Kota Mojokerto (Sumber : Web Site Kota Mojokerto)

No	Kecamatan	Luas Wilayah (Ha)	Jumlah			
			Kelurahan	Dusun	RW	RT
	Kecamatan Prajurit Kulon	776,267	8	31	58	236
1	Surodinawan	145,875	1	4	6	23
2	Kranggan	113,307	1	5	12	38
3	Miji	39,600	1	3	11	47
4	Blooto	178,065	1	4	9	30
5	Prajurit Kulon	119,500	1	3	5	25
6	Mentikan	18,900	1	4	4	23
7	Kauman	18,635	1	3	3	21
8	Pulorejo	142,355	1	5	8	29
	Kecamatan Magersari	870,210	10	38	92	325
1	Meri	164,841	1	3	8	26
2	Gunung Gedangan	170,455	1	6	7	25
3	Kedundung	228,575	1	5	10	37
4	Balongsari	82,860	1	4	14	46
5	Jagalan	16,555	1	2	6	18
6	Sentanan	13,850	1	2	6	18
7	Purwotengah	13,557	1	3	3	20
8	Gedongan	14,530	1	2	4	14
9	Magersari	32,892	1	4	10	35
10	Wates	132,095	1	7	24	86

zona wilayah. Pembagian wilayah Mojokerto ke dalam zona-zona dapat dilihat pada tabel 3.3.

Tabel 3.2 Rute Trayek Angkutan Kota di Kota Mojokerto (Laporan Umum PKL^[11])

Trayek	Rute
A	Terminal Kertajaya-Jl Jayanegara-Jl Pahlawan-Jl Gajah Mada-Jl R Pamuji-Jl.Letkol Sumarjo- Jl Hayam Wuruk-Jl Gajah Mada-Jl Trunojoyo-Jl Ronggolawe-Jl Raya Ijen-Jl Semeru- Jl Penanggungan-Jl Batok- Jl Raya Ijen- Jl Raya Ijen-Jl Batok-Jl Penanggungan-Jl.Semeru-Jl Raya Ijen-Jl Ronggolawe-Jl Trunojoyo- Jl Gajah Mada-Jl Taman Siswa- Jl Sersan Harun-Jl R Pamuji-Jl Gajah Mada-Jl Pahlawan- Jl Jayanegara- Terminal Kertajaya
B	Terminal Kertajaya-Jl Jayanegara- Jl R Wijaya- Jl.Majapahit -Jl R A Basuni- Jl K H Usman- Jl.Prajurit Kulon-Jl Tri Buanan Tungga Dewi- Jl Brawijaya-Jl Hayam Wuruk- Jl.Majapahit -Jl A Yani-Jl Letkol Sumarjo-Jl R Pamuji-Jl.Gajah Mada-Jl Pahlawan-Jl Tropodo-Jl Meri- Terminal Kertajaya
C	Terminal Kertajaya-Jl Jayanegara- Jl Wijaya Kusuma-Jl RA Basuni- <i>Sooko- Brangkal – Sooko</i> -Jl RA Basuni -Jl Brawijaya-Jl KH Wachid Hasyim- Jl Bhayangkara-Jl KH Nawawi- Jl Cokroaminoto-Jl Gajah Mada-Jl Pahlawan Jl Jayanegara-Terminal Kertajaya
D	Terminal Kertajaya-Jl Meri-Jl Tropodo- Jl Pahlawan -Jl Gajah Mada -Jl Cokroaminoto- Jl KH Nawawi- Jl Bayangkara-Jl.Majapahit-Jl Brawijaya-Jl Tri Buanan Tungga Dewi- Jl.Prajurit Kulon - <i>Sambiroto-Brangkal-Sambiroto</i> - Jl.KH Usman-Jl.Wijaya Kusuma-Jl Teratai-Jl R Wijaya-Jl Jaya Negara- Terminal Kertajaya
E	<i>Jabon- Gayaman – Kepuh Anyar</i> - Jl Sekar Putih- Jl Empunala- Jl R Pamuji-Jl Letkol Sumarjo -Jl WR Supratman-Jl Majapahit- Jl Yos Sudarso-Jl Prapanca-Jl Pulorejo Wetan- <i>Ngingas Rembyong</i> - Jl Pulorejo Wetan -Jl Brawijaya-Jl Majapahit-Jl A Yani-Jl Letkol sumarjo-Jl Taman Siswa- Jl Gajah Mada -Jl Empu Nala-Jl Sekar Putih- <i>Kepuh Anyar- Gayaman – Jabon</i>
F	Terminal Kertajaya-Jl By Pass- Jl Empunala-Jl Gajah Mada- <i>Lespadangan</i> - Jl Gajah Mada - Jl Empunala- Jl By Pass -

Tabel 3.3 Pembagian Wilayah Mojokerto Dalam Zona (Laporan Umum PKL)

ZONA	DESKRIPSI	KELURAHAN / DESA
1	GRID	Sentanan
		Jagalan
		Purwotengah
2	GRID	Purwotengah
		Kauman
		Mentikan
3	BATAS ADMINISTRASI	Magersari
		Gedongan
4	BATAS ADMINISTRASI	Balongsari
5	GRID	Kranggan
		Miji
6	GRID	Mentikan
		Miji
7	BATAS ADMINISTRASI	Wates
8	BATAS ADMINISTRASI	Kedundung
9	BATAS ADMINISTRASI	Gunung gedangan
10	BATAS ADMINISTRASI	Meri
11	BATAS ADMINISTRASI	Banjar Agung
12	BATAS ADMINISTRASI	Sooko
13	BATAS ADMINISTRASI	Surodinawan
14	BATAS ADMINISTRASI	Blooto
15	GRID	Prajurit Kulon
		Mentikan

3. 1. 1. 2. **Kondisi Sistem Perencanaan Angkutan Kota Pada Saat Ini**

Selama ini Dinas Perhubungan Kota Mojokerto melakukan evaluasi terhadap kinerja pelayanan angkutan kota setiap tahun, untuk dilihat apakah kinerja pelayanan angkutan kota yang ada masih baik, perlu dilakukan perencanaan ulang dan ataukah perlu dilakukannya penambahan jumlah armada baru. Tetapi pembahasan hanya dibatasi pada perencanaan ulang trayek angkutan kota saja, sedangkan besarnya jumlah armada tidak dicari. Dimana perencanaan ulang di sini adalah penataan ulang jalur atau trayek angkutan kota sehingga akan menghasilkan jalur trayek angkutan kota yang baru, yang diharapkan trayek tersebut akan diminati oleh para pengguna jasa angkutan kota.

Selama ini dalam perencanaan trayek angkutan kota yang dilakukan untuk perencanaan pada suatu tahun mendatang sebagian prosesnya masih dilakukan secara manual, dan sebagian proses lainnya sudah menggunakan bantuan aplikasi tertentu. Dalam hal ini dengan menggunakan Tranplan yang merupakan sebuah aplikasi dalam bidang transportasi dan SPSS yang merupakan sebuah aplikasi untuk melakukan peramalan. Peramalan disini digunakan untuk mengetahui jumlah bangkitan perjalanan yang terjadi pada tahun yang direncanakan. Tetapi tidak ada aplikasi khusus yang digunakan untuk melakukan perencanaan trayek angkutan kota.

Selain itu selama ini pula semua data-data hasil survey yang digunakan untuk melakukan perencanaan ulang trayek angkutan kota tidak tersimpan dengan baik, data-data tersebut hanya ditulis dalam bentuk dokumen di atas kertas, tidak tersimpan dalam sebuah basis-data. Hal ini dapat menyebabkan terjadinya kemungkinan data tersebut terselip, atau kesulitan untuk mencari data yang

sebelumnya. Pengertian baik disini adalah sebuah sistem yang lebih cepat, dan mudah untuk digunakan, bila dibandingkan dengan sistem sebelumnya. Selain itu diharapkan pula sistem tersebut merupakan sebuah sistem yang telah terintegrasi prosesnya dari awal sampai dengan akhir, dengan menggunakan suatu aplikasi khusus perencanaan ulang rute trayek angkutan kota. Diperlukan sebuah sistem baru untuk perencanaan ulang trayek angkutan kota dengan masukan awal untuk sistem tersebut berupa hasil-hasil survey yang dilakukan oleh Dinas Perhubungan Kota Mojokerto, dan diharapkan menghasilkan beberapa luaran sebagai berikut:

- Rancangan jalur atau rute trayek baru sesuai dengan kondisi yang ada,
- Evaluasi jalur atau rute trayek angkutan kota baru, apakah mempunyai rasio waktu perjalanan dan rasio biaya perjalanan yang rendah. Dimana kedua rasio itu diperoleh dengan melakukan perbandingan antara penggunaan angkutan kota dengan menggunakan kendaraan pribadi,
- Peluang dan pangsa pasar rute baru trayek angkutan kota, apakah peluang penggunaan rute baru trayek angkutan kota lebih tinggi dari rute lamanya atau tidak pada tahun sekarang dan tahun rencana.

Kemudian diharapkan pula, dari hasil survey yang digunakan sebagai masukan aplikasi, dan hasil dari perencanaan yang dilakukan dapat tersimpan dengan baik dalam suatu basis-data. Sehingga akan memudahkan dalam melakukan penyimpanan data, tidak perlu khawatir data tersebut terselip, mudah dalam melakukan pencarian data yang diinginkan, dan tidak memakan tempat untuk penyimpanannya.

3. 1. 3. Deskripsi Kebutuhan Sistem

Sub-bab ini menjelaskan bagaimana memenuhi keadaan sistem seperti

membuat perencanaan ulang jalur trayek angkutan kota, dimana dalam pembuatannya berdasarkan pada ilmu transportasi yang telah ada selama ini untuk perencanaan jalur trayek angkutan kota.

Tabel 3.4 Perbandingan antara kondisi sistem saat ini dan yang diinginkan

No	Kondisi Sistem Saat Ini	Kondisi Sistem Yang Diinginkan
1	Tidak terintegrasi, sebagian proses dilakukan secara manual, dan sebagian menggunakan beberapa aplikasi pendukung. Tidak ada aplikasi khusus perencanaan trayek angkutan kota.	Proses yang ada diharapkan sudah terintegrasi dalam satu aplikasi khusus perencanaan trayek angkutan kota. <i>Dengan masukan:</i> data survey yang dilakukan pihak Dinas Perhubungan. <i>Dan luaran:</i> Rute trayek angkutan kota rencana, dan evaluasi rute baru tersebut.
2	Proses yang dilakukan menjadi lebih lama dan sulit karena proses yang ada tersebut tidak terintegrasi	Proses yang ada menjadi lebih mudah dan cepat, karena pengguna hanya perlu memasukkan data hasil survey, dan selanjutnya semua proses yang ada dijalankan oleh aplikasi
3	Aplikasi yang digunakan selama ini tidak <i>user friendly</i> , sulit dalam penggunaannya.	Aplikasi yang <i>user friendly</i> , dan dapat digunakan oleh siapapun, tanpa perlu keahlian khusus
4	Data hasil survey dan hasil perencanaan tidak tersimpan dalam suatu Basis-data	Data hasil survey dan hasil perencanaan tersimpan dalam suatu Basis-data, sehingga akan memudahkan bila sewaktu-waktu data tersebut dibutuhkan nantinya
5	Pihak berwenang harus melakukan analisis sendiri untuk perencanaan ulang trayek angkutan kota	Pihak berwenang lebih terbantu, karena telah diberikan analisis terhadap perencanaan ulang trayek angkutan kota. Sehingga berwenang mengetahui rute trayek apa saja yang harus dirubah dan apakah rute tersebut berpengaruh terhadap pangsa pasar angkutan

Data yang digunakan untuk pengaplikasian aplikasi perencanaan trayek angkutan kota hanya dikhususkan pada kota Mojokerto, dengan data hasil survey yang dilakukan oleh pihak Dinas Perhubungan Kota Mojokerto. Selain itu dalam pembuatan aplikasi untuk perencanaan trayek angkutan kota di kota Mojokerto juga disesuaikan dengan kondisi di wilayah Mojokerto tersebut, karakteristik sosial ekonomi penduduknya, dan juga karakteristik angkutan kota yang telah ada sebelumnya (*eksisting*) yang selanjutnya akan dilakukan perencanaan ulang berdasarkan pada kondisi tersebut.

Tujuan dari pembuatan aplikasi untuk perencanaan trayek angkutan kota seperti telah dijelaskan di atas dan sesuai dengan kondisi sistem yang diinginkan seperti telah disebutkan pada sub-bab 3.1.2 adalah untuk mempermudah dalam melakukan proses perencanaan. Aplikasi ini hanya membantu dalam memberikan perencanaan ulang trayek angkutan kota, sedangkan implementasi nyata di lapangan hanya tergantung pada pihak yang berwenang.

3.2. PERANCANGAN DATA

Sub-bab ini menjelaskan tentang perancangan data dari aplikasi untuk perencanaan trayek angkutan kota. Perancangan data tersebut dimulai dengan pembuatan perancangan data secara konseptual dengan menggunakan *entity relational modelling* yang akan menghasilkan *Entity Relational Database* (ERD). Untuk menggambarkan model perancangan tersebut, simbol-simbol yang digunakan mengacu pada notasi El Masri^[4]. Setelah itu dilanjutkan dengan perancangan model data secara logikal, yang dilakukan dengan melakukan pemetaan terhadap ERD ke dalam bentuk tabel. Langkah terakhir adalah



Tipe	Keterangan
Atribut	Atribut node_pusat merupakan atribut yang menandakan bahwa node tersebut termasuk node pusat bila bernilai 1 dan bukan node pusat bila bernilai 0.
Entitas	Entitas link merupakan sebuah entitas yang berisi data link-link di kota Mojokerto, yang terhubung oleh dua buah node. Beberapa link akan membentuk jalan, dan kesatuan seluruh link akan membentuk sebuah jaringan jalan. Entitas ini mempunyai enam atribut, dapat dilihat pada point 4.1 dan 4.6.
Atribut	Atribut id_link merupakan atribut kode, yang akan membedakan link yang satu dengan link yang lain di kota Mojokerto.
Atribut	Atribut nama_link merupakan atribut nama untuk tiap link, yang terdiri dari dua bagian, yaitu nama node awal dan nama node akhir. Misalnya 0101-0102, yang berarti bahwa link ini menghubungkan antara node 0101 dengan 0102.
Atribut	Atribut panjang merupakan atribut yang berisi data panjang tiap link yang ada di kota Mojokerto.
Atribut	Atribut arus_lalu_lintas merupakan atribut yang berisi data arus lalu lintas tiap link yang ada di kota Mojokerto, banyaknya kendaraan yang melewati link tertentu tiap jamnya dengan satuan smp/jam (satuan mobil penumpang per jam)
Atribut	Atribut kapasitas merupakan atribut yang berisi data kapasitas tiap link yang ada di kota Mojokerto. Link tersebut dapat menampung berapa banyak kendaraan.
Atribut	Atribut kecepatan merupakan atribut yang berisi data rata-rata kecepatan kendaraan pada tiap link yang ada di kota Mojokerto.
Entitas	Entitas sampel_HI merupakan entitas yang berisi tentang jumlah seluruh keluarga yang ada di tiap zona, jumlah keluarga bersedia diwawancara, jumlah rumah yang kosong, dan berapa banyak keluarga yang menolak untuk diwawancara. Sampel survey ini berfungsi untuk mencari faktor ekspansi atau faktor perluasan data sampel survey HI.

	Tipe	Keterangan
arga	Atribut	Atribut jumlah_seluruh_keluarga adalah atribut banyaknya rumah tangga atau keluarga pada tiap zona dalam wilayah kota Mojokerto.
g	Atribut	Atribut jumlah_rumah_kosong adalah atribut banyaknya rumah yang kosong pada tiap zona dalam wilayah kota Mojokerto.
wawancara	Atribut	Atribut keluarga_menolak_diwawancara adalah atribut banyaknya rumah tangga atau keluarga yang menolak untuk diwawancarai pada tiap zona dalam wilayah kota Mojokerto.
rvey	Atribut	Atribut jumlah_keluarga_disurvey adalah atribut banyaknya keluarga yang mau diwawancara pada tiap zona dalam wilayah kota Mojokerto
	Atribut	Merupakan atribut yang mencatat tahun pendataan jumlah data sampel HI dilakukan
	Entitas Lemah	Entitas Total_Data_Keluarga merupakan entitas lemah dari entitas zona, artinya entitas Total_Data_Keluarga merupakan bagian dari atau dimiliki oleh entitas zona. Satu Total_Data_Keluarga hanya dimiliki oleh satu zona, dan tidak lebih. Entitas Total_Data_Keluarga di sini merupakan jumlah total hasil survey HI tiap keluarga yang dilakukan oleh pihak Dinas perhubungan kota Mojokerto.
ga	Atribut	Atribut ini merupakan atribut kode yang membedakan antara total data keluarga yang satu dengan total data keluarga yang lain
ga	Atribut	Atribut ini merupakan atribut total jumlah anggota seluruh keluarga yang disurvey.
	Atribut	Atribut ini merupakan atribut total pendapatan untuk seluruh keluarga yang disurvey.
	Atribut	Atribut ini merupakan jumlah total kendaraan pribadi seluruh keluarga yang disurvey
	Atribut	Atribut ini merupakan atribut yang mencatat tahun pendataan survey tiap keluarga dilakukan

	Tipe	Keterangan
	Entitas	Merupakan entitas yang berisi data kelurahan
	Atribut	Merupakan atribut kode, yang membedakan antara kelurahan yang satu dengan yang lain
	Atribut	Merupakan atribut yang berisi nama-nama kelurahan
	Relasi	Relasi ini merupakan relasi pergerakan orang dari satu zona ke zona lain yang ada di Kota Mojokerto
	Atribut	Atribut jumlah_perjalanan merupakan atribut banyaknya pergerakan atau perjalanan yang dilakukan orang dari satu zona ke zona lain. Jumlah perjalanan di sini adalah total banyaknya perjalanan yang dilakukan oleh penduduk Kota Mojokerto.
angkutan	Atribut	Atribut jumlah_perjalanan_angkutan merupakan atribut banyaknya pergerakan atau perjalanan yang dilakukan orang dengan menggunakan angkutan kota dari satu zona ke zona lain.
	Atribut	Atribut ini merupakan atribut yang mencatat tahun pendataan survey pergerakan orang dilakukan
	Relasi	Atribut ini merupakan atribut relasi. Relasi ini menyebutkan setiap kelurahan yang ada di kota Mojokerto termasuk pada wilayah zona apa. Satu zona dapat terdiri dari beberapa kelurahan, dan satu kelurahan dapat berada pada dua zona yang berbeda

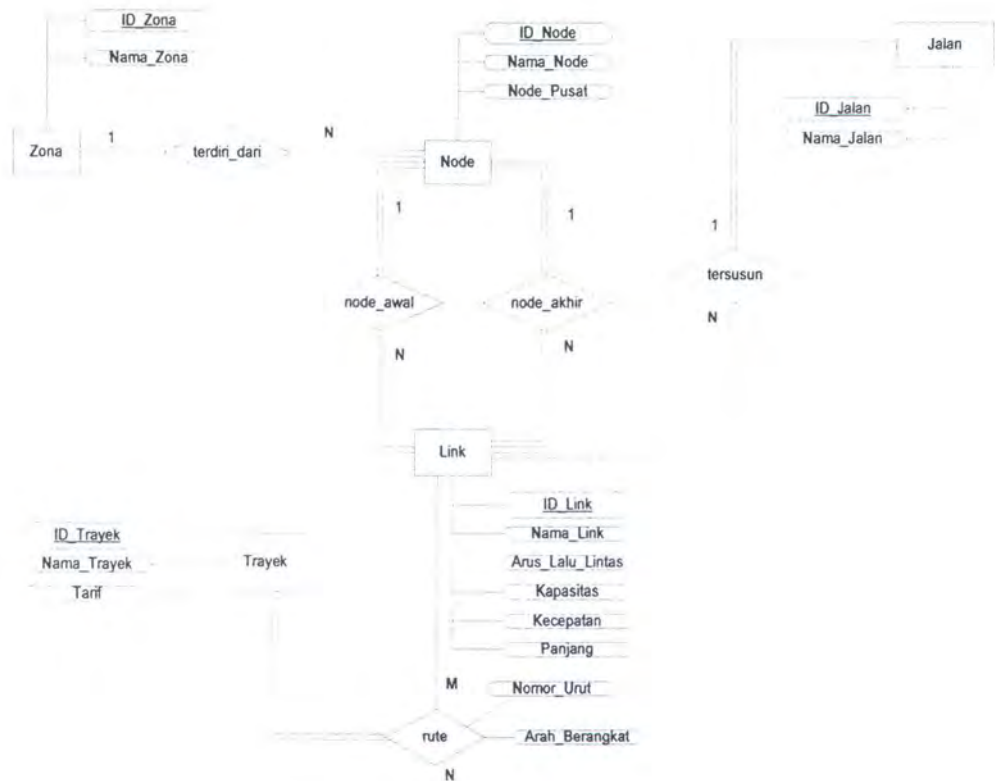
survey, yaitu entitas jalan, link, node, dan zona, yang digunakan bersama-sama. Penjelasan dari masing-masing entitas tersebut beserta atributnya dapat dilihat pada tabel 3.6.

Secara garis besar relasi antara entitas pada pemodelan data trayek seperti ditunjukkan dalam gambar 3.3 adalah relasi rute yang merupakan relasi menghubungkan antara entitas trayek dengan entitas link, dengan hubungan relasional banyak-ke-banyak yang berarti bahwa tiap trayek mempunyai rute lebih dari satu link, dan satu link dapat dilewati oleh rute lebih dari satu trayek.

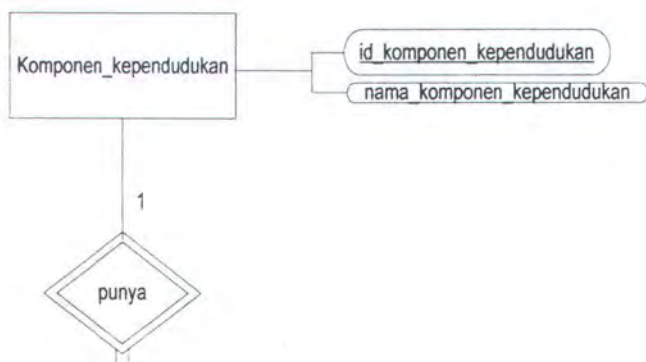
2.1.3. ERD Data Kependudukan

Data kependudukan merupakan data-data yang berisi tentang hal-hal yang berhubungan dengan faktor kependudukan di Kota Mojokerto untuk waktu beberapa tahun, dimana data ini nantinya digunakan dalam proses peramalan angkutan perjalanan. Pada ERD data kependudukan ini hanya terdapat 1 entitas kuat yaitu komponen kependudukan yang merupakan sebuah entitas yang berisi jenis data-data yang berhubungan dengan faktor-faktor kependudukan Kota Mojokerto, dan 1 entitas lemah item komponen kependudukan yang merupakan sebuah entitas yang berisi penjabaran lebih detail dari jenis data pada entitas komponen_kependudukan serta nilai dari masing-masing komponen kependudukan tersebut.

Secara garis besar relasi antara entitas ERD data kependudukan adalah relasi punya yang mempunyai hubungan relasional satu-ke-banyak dan bersifat



Gambar 3.3 Pemodelan data trayek



Tabel 3.6 Deskripsi Tiap Entitas dan Atribut Pemodelan Data Trayek

Nama	Tipe	Keterangan
	Entitas	Entitas trayek merupakan entitas yang berisi data-data trayek angkutan kota yang ada di kota Mojokerto. Entitas ini mempunyai tiga atribut, yang dapat dilihat pada point 5.1 sampai 5.3.
	Atribut	Atribut id_trayek merupakan atribut kode, yang akan membedakan trayek yang satu dengan trayek yang lain di kota Mojokerto.
k	Atribut	Atribut nama_trayek merupakan atribut nama-nama trayek yang ada di kota Mojokerto.
	Atribut	Merupakan atribut tarif penumpang untuk tiap trayek yang ada di Kota Mojokerto
	Relasi	Merupakan relasi tentang link-link yang dilewati oleh suatu trayek tertentu, yang nantinya akan terbentuk suatu jalur atau rute dari trayek tersebut.
	Atribut	Merupakan atribut yang memberikan penomoran untuk link yang dilewati oleh suatu trayek, misalnya nomor urut 1 untuk trayek A melewati link 0101-0102, nomor urut 2 untuk trayek A melewati link 0102-0103, nomor urut 3 untuk trayek A melewati link 0103-0104, hal ini berarti rute trayek A adalah (0101-0102) – (0102-0103) – (0103-0104)
kat	Atribut	Merupakan atribut yang menyatakan arah berangkat dari rute angkutan kota untuk nilai 1, dan arah pulang untuk nilai 0.

Tabel 3.7 Deskripsi Tiap Entitas dan Atribut Pemodelan Data Kependudukan

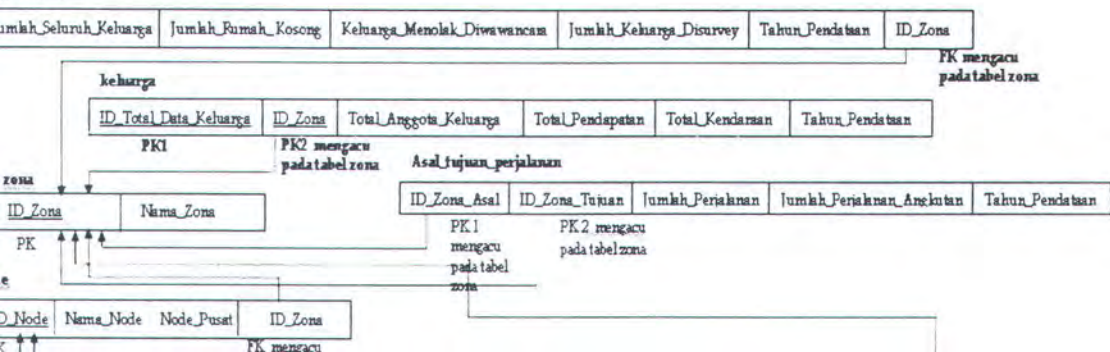
Nama	Tipe	Keterangan
pendudukan	Entitas	Merupakan entitas yang berisi pengelompokkan data-data untuk komponen kependudukan., seperti jumlah pendapatan penduduk, kepemilikan kendaraan, dan lain sebagainya
_Kependudukan	Atribut	Atribut id_komponen_kependudukan merupakan atribut kode, yang akan membedakan komponen kependudukan yang satu dengan komponen kependudukan yang lain di kota Mojokerto.
en	Atribut	Merupakan atribut jenis komponen kependudukan
pendudukan	Entitas	Merupakan entitas yang berisi data item-item yang termasuk dalam penggolongan tiap komponen kependudukan
omponen_Kependudukan	Atribut	Atribut id_item_komponen_kependudukan merupakan atribut kode, yang akan membedakan item_komponen kependudukan yang satu dengan item_komponen kependudukan yang lain di kota Mojokerto.
	Atribut	Merupakan tahun untuk tiap data item komponen kependudukan
	Atribut	Merupakan nilai item komponen kependudukan untuk suatu tahun tertentu pada suatu zona. Misalnya komponen kependudukan “jumlah penduduk” untuk tahun 2000 pada zona 1 mempunyai nilai 1000.

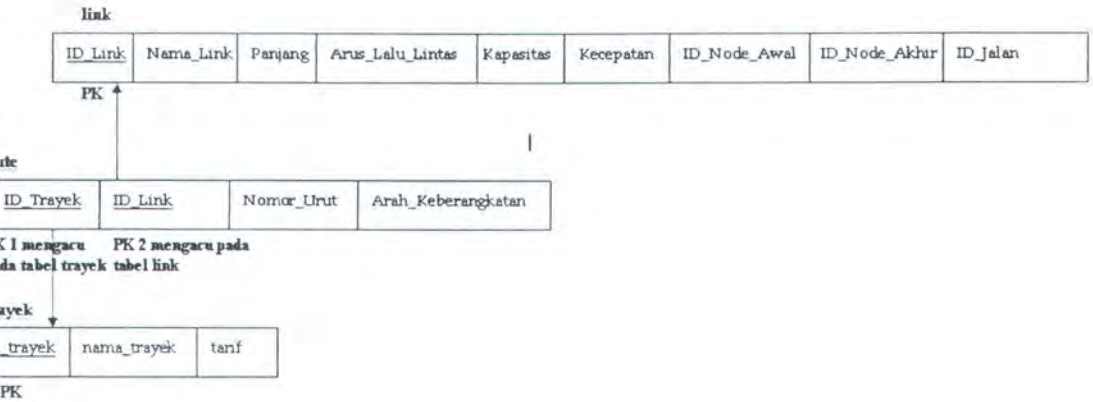
2. Pemetaan ERD Menjadi Tabel

Tahap selanjutnya dalam perancangan data adalah pemetaan ERD ke dalam bentuk tabel-tabel. Pada tahapan ini akan terbentuk primary key dan juga foreign key untuk masing-masing tabel yang terbentuk dari tiap hubungan fungsional antara entitas yang satu dengan entitas yang lain pada model ERD. Terdapat 3 ERD yang akan dipetakan ke dalam bentuk skema relasi, yaitu pemetaan ERD data survey, ERD data trayek, dan ERD data kependudukan.

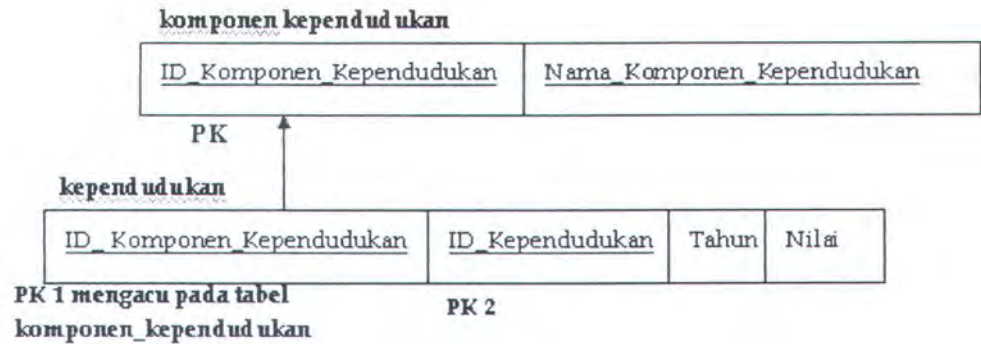
Pemetaan ERD data survey dapat dilihat pada gambar 3.5, pada hasil pemetaan tersebut terbentuk 2 tabel baru, yaitu tabel wilayah_zona dan tabel asal_tujuan_perjalanan. Pemetaan ERD data trayek dapat dilihat pada gambar 3.6. Pada pemetaan tersebut terbentuk satu buah tabel baru, yaitu tabel rute, yang merupakan hasil hubungan relational banyak-ke-banyak rute antara entitas trayek dengan entitas link. Dan yang terakhir, pemetaan ERD data kependudukan dapat dilihat pada gambar 3.7.

Contoh untuk HI





Gambar 3.6 Pemetaan ERD Data Trayek

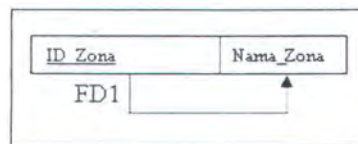


Gambar 3.7 Pemetaan ERD Data Kependudukan

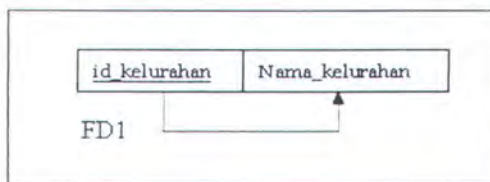
3. Normalisasi Data

Tahap ketiga dari perancangan data adalah melakukan normalisasi terhadap tabel yang sudah ditetapkan, seperti telah dijelaskan pada sub-bab 3.2.2.

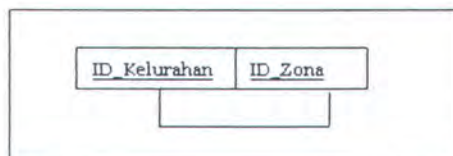
lakukan pada setiap tabel hasil pemetaan. Tabel-tabel tersebut adalah tabel zona, tabel node, tabel link, tabel jalan, tabel sampel_HI, tabel_kelurahan, tabel_tal_data_keluarga, tabel asal_tujuan_perjalanan, tabel_wilayah_zona, tabel_ayek, tabel rute, tabel komponen kependudukan, tabel item komponen kependudukan. FD masing-masing tabel tersebut dapat dilihat pada gambar 3.8 sampai dengan gambar 3.20



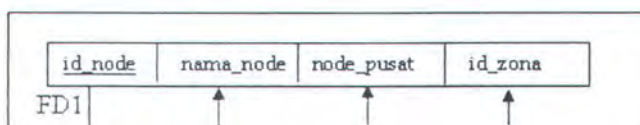
Gambar 3.8 Ketergantungan Fungsional Tabel Zona

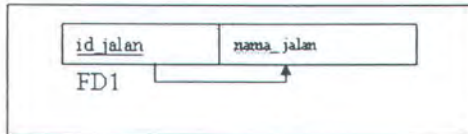


Gambar 3.9 Ketergantungan Fungsional Tabel Kelurahan

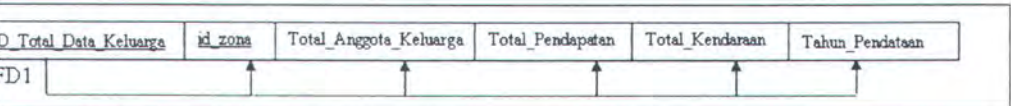


Gambar 3.10 Ketergantungan Fungsional Tabel Wilayah Zona

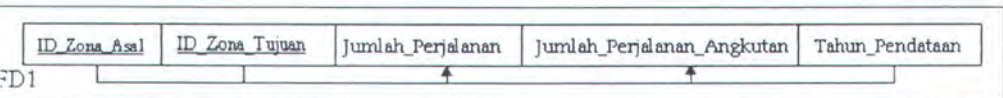




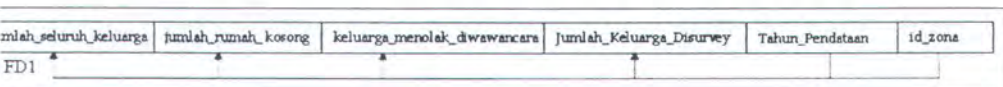
Gambar 3.13 Ketergantungan Fungsional Tabel Jalan



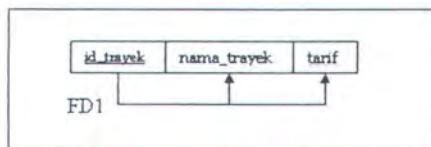
Gambar 3.14 Ketergantungan Fungsional Tabel Total Data Keluarga



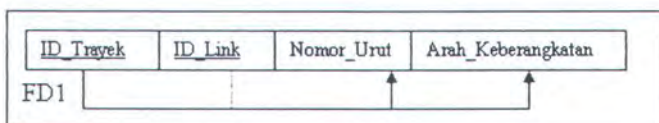
Gambar 3.15 Ketergantungan Fungsional Tabel Asal Tujuan Perjalanan

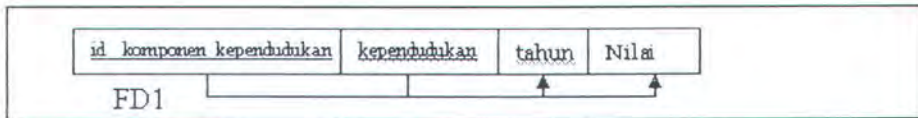


Gambar 3.16 Ketergantungan Fungsional Tabel Sampel HI



Gambar 3.17 Ketergantungan Fungsional Tabel Trayek





Gambar 3.20 Ketergantungan Fungsional Tabel Kependudukan

2. 3. 2. Normalisasi 2 NF

Dikatakan normalisasi 2 NF bila seluruh atribut non key bergantung sepenuhnya pada atribut key. Dari FD pada gambar 3.8 sampai dengan 3.20 dapat dilihat bahwa seluruh atribut dari seluruh tabel yang ada bergantung sepenuhnya pada atribut key, hal ini berarti seluruh tabel tersebut telah normalisasi 2 NF.

2. 3. 3. Normalisasi 3 NF

Dikatakan normalisasi 3 NF bila tidak ada yang transitif dan telah 2 NF. Tabel transitif terjadi bila ada $a \rightarrow b \rightarrow c$ maka $a \rightarrow c$. Functional dependency pada gambar 3.8 sampai dengan 3.20 di atas dapat dilihat tidak ada yang transitif dan seluruhnya telah normalisasi 2 NF, hal ini berarti bahwa seluruh tabel tersebut telah normalisasi 3 NF.

2. 3. 4. Tabel Setelah Normalisasi

Seluruh tabel hasil pemetaan dari 3 ERD, seperti yang telah dijelaskan pada sub-bab 3.2.2, sudah dikatakan memenuhi syarat normalisasi 2 NF dan 3 NF, dan berikut deskripsi tabel untuk lebih detailnya, dapat dilihat pada tabel 3.8 sampai dengan tabel 3.20

Tabel 3.15 Tabel Total Data Keluarga

Nama Atribut	Keterangan
ID_Total_Data_Keluarga	PK1
ID_Zona	PK2 mengacu pada tabel zona (id_zona)
Total_Anggota_Keluarga	
Total_Pendapatan	
Total_Kendaraan	
Tahun_Pendataan	

Tabel 3.16 Tabel Sampel HI

Nama Atribut	Keterangan
Jumlah_Seluruh_Keluarga	
Keluarga_Menolak_Diwawancara	
Jumlah_Rumah_Kosong	
Jumlah_Keluarga_Disurvey	
ID_Zona	FK mengacu pada tabel zona (id_zona)
Tahun_Pendataan	

Tabel 3.17 Tabel Trayek

Nama Atribut	Keterangan
ID_Trayek	PK
Nama_Trayek	
Tarif	

Tabel 3.18Tabel Rute

Nama Atribut	Keterangan
ID_Trayek	PK1 mengacu pada tabel trayek
ID_Zona	PK2 mengacu pada tabel zona

Tabel 3.20 Tabel Kependudukan

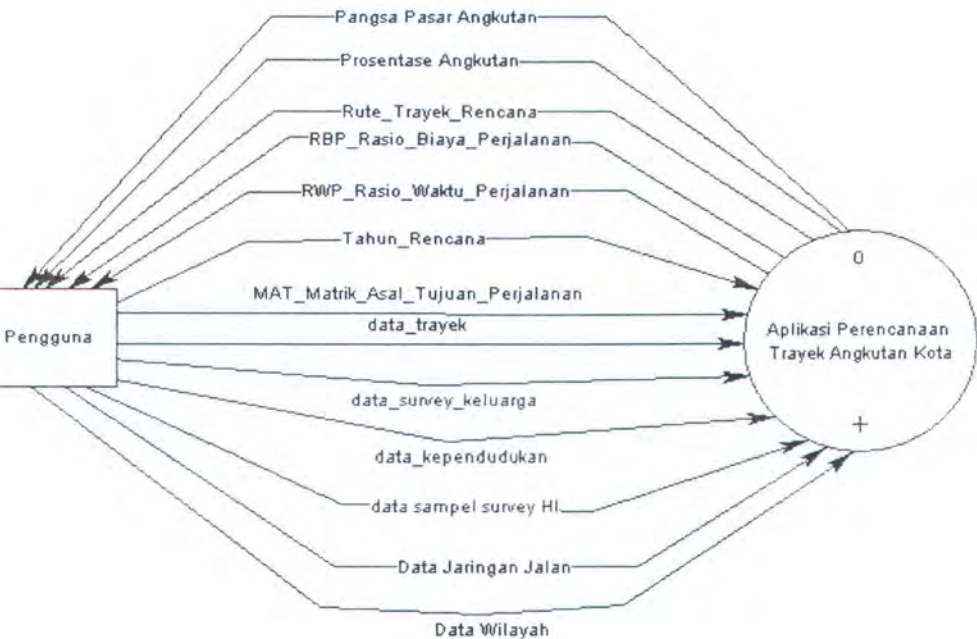
Nama Atribut	Keterangan
_Komponen_Kependudukan	PK1 mengacu pada tabel komponen_kependudukan
_Kependudukan	PK2
nun	
ai	

3. PERANCANGAN PROSES

Untuk menggambarkan proses perencanaan trayek angkutan kota dilakukan secara terstruktur dari proses secara umum sampai dengan proses lebih detailnya. Proses tersebut disajikan dalam bentuk Diagram Kontek Data untuk proses paling umum, dan Diagram Alir Data (*Data Flow Diagram/DFD*) untuk proses-proses detailnya. Untuk proses paling akhir yang sudah tidak dapat dipecah menjadi proses lain, digambarkan dengan menggunakan *flowchart* sebelum diubah ke dalam bentuk *coding*. Untuk menggambarkan proses DFD ini digunakan notasi pada Aplikasi Power Designer 9.

3.1. DFD Level 0 (Diagram Kontek Data)

Pada DFD level 0 ini Perencanaan Trayek Angkutan Kota (PTAK) menerima masukan dari pengguna berupa data trayek, data survey keluarga, data sampel survey HI, data kependudukan, data MAT (Matrik Asal Tujuan) perjalanan, Data Jaringan Jalan, Data Wilayah, dan masukan nilai tahun trayek tersebut akan direncanakan. Kemudian data-data tersebut diolah sehingga



Gambar 3.21 Diagram Kontek Data Aplikasi Perencanaan Trayek Angkutan Kota

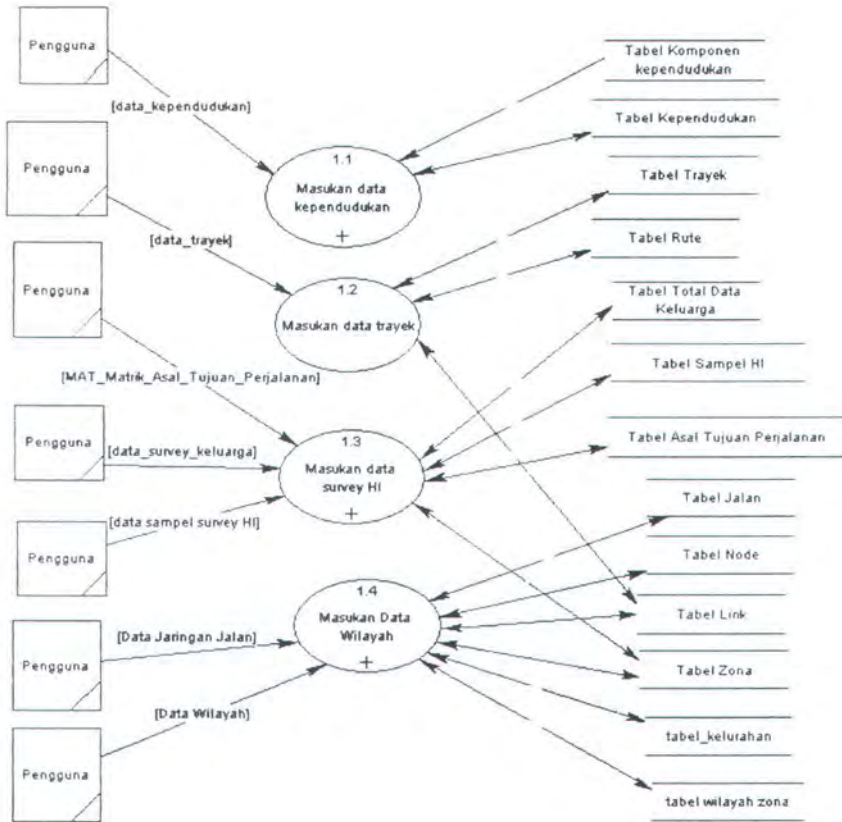
3.2. Aplikasi Perencanaan Trayek Angkutan Kota

“Aplikasi Perencanaan Trayek Angkutan Kota” merupakan DFD level 1 (lihat pada gambar 3.22) yang mempunyai 2 sub proses di dalamnya, yaitu:

Proses 1 : Masukan Basis-data

Proses 2 : Proses Perencanaan Trayek

3.2.1. Masukan Basis-Data



Gambar 3.23 DFD Level 2 Masukan Basis-Data

3.2 1.1 Proses Masukan Data Kependudukan

Proses ini adalah proses untuk memasukkan data kependudukan untuk beberapa tahun. Dimana data kependudukan tersebut selanjutnya dimasukkan ke dalam tabel kependudukan, dengan kode yang berbeda-beda untuk tiap jenis data kependudukan. Kode data kependudukan tersebut diambil dari tabel komponen

ID_Komponen_Kependudukan = 3, merupakan kode untuk data jumlah kendaraan

Proses masukan data kependudukan merupakan DFD level 3 yang mempunyai 3 proses di dalamnya, yaitu:

Proses 1.1.1 : Masukan Data Jumlah Penduduk.

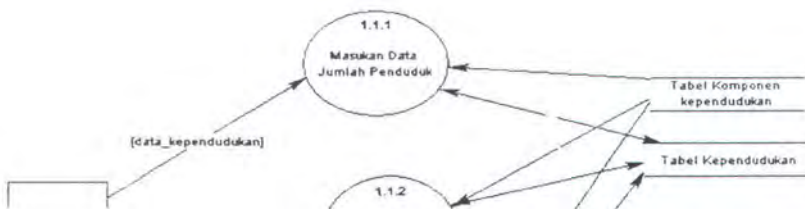
Proses 1.1.2 : Masukan Data Jumlah Pendapatan.

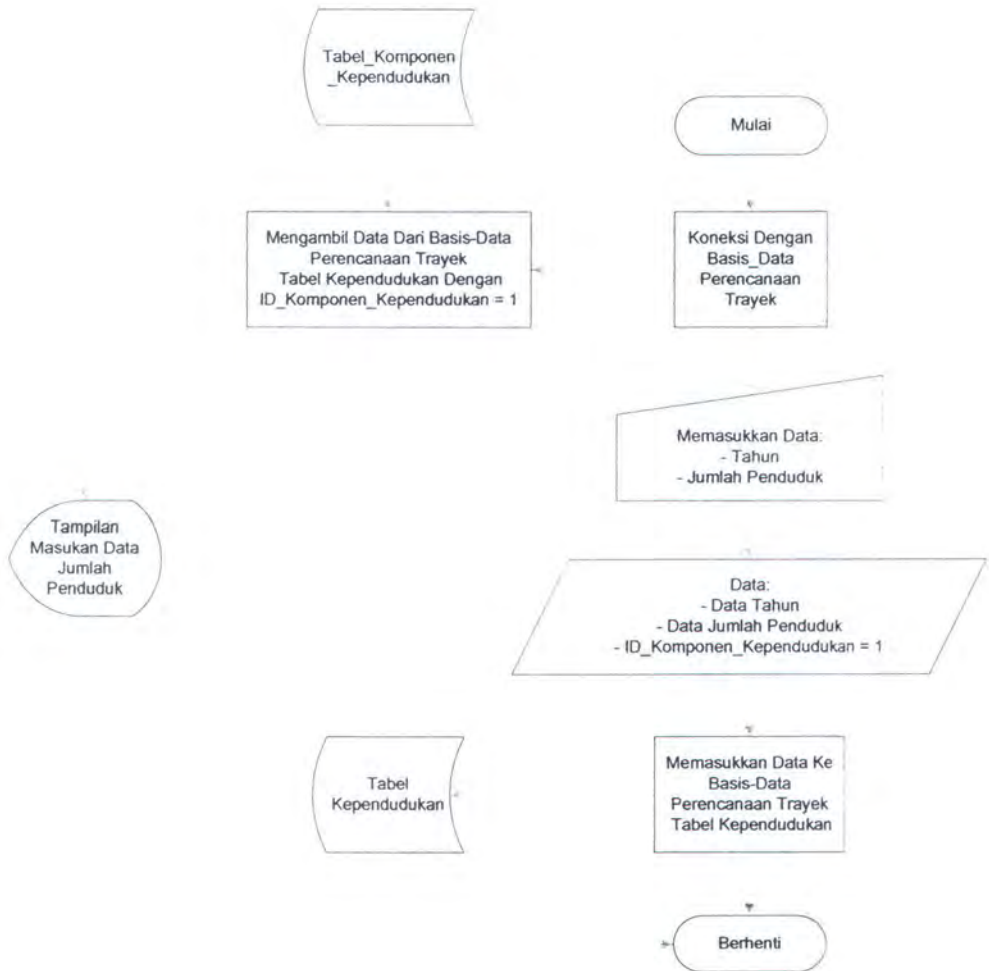
Proses 1.1.3 : Masukan Data Jumlah Kendaraan.

Selanjutnya DFD Level 3 proses masukan data kependudukan tersebut dapat dilihat pada gambar 3.24.

3. 2. 1. 1. 1. Proses Masukan Data Jumlah Penduduk

Proses ini adalah proses untuk memasukkan data jumlah penduduk untuk periode beberapa tahun. Hasil dari proses ini nantinya akan dimasukkan ke dalam tabel kependudukan dengan ID_Komponen_Kependudukan pada tabel komponen_kenpendudukan sama dengan satu. Data-data yang dimasukkan dalam proses ini adalah tahun dan jumlah penduduk pada tahun yang dipilih tersebut. Proses detailnya dapat dilihat pada *flowchart* gambar 3.25.



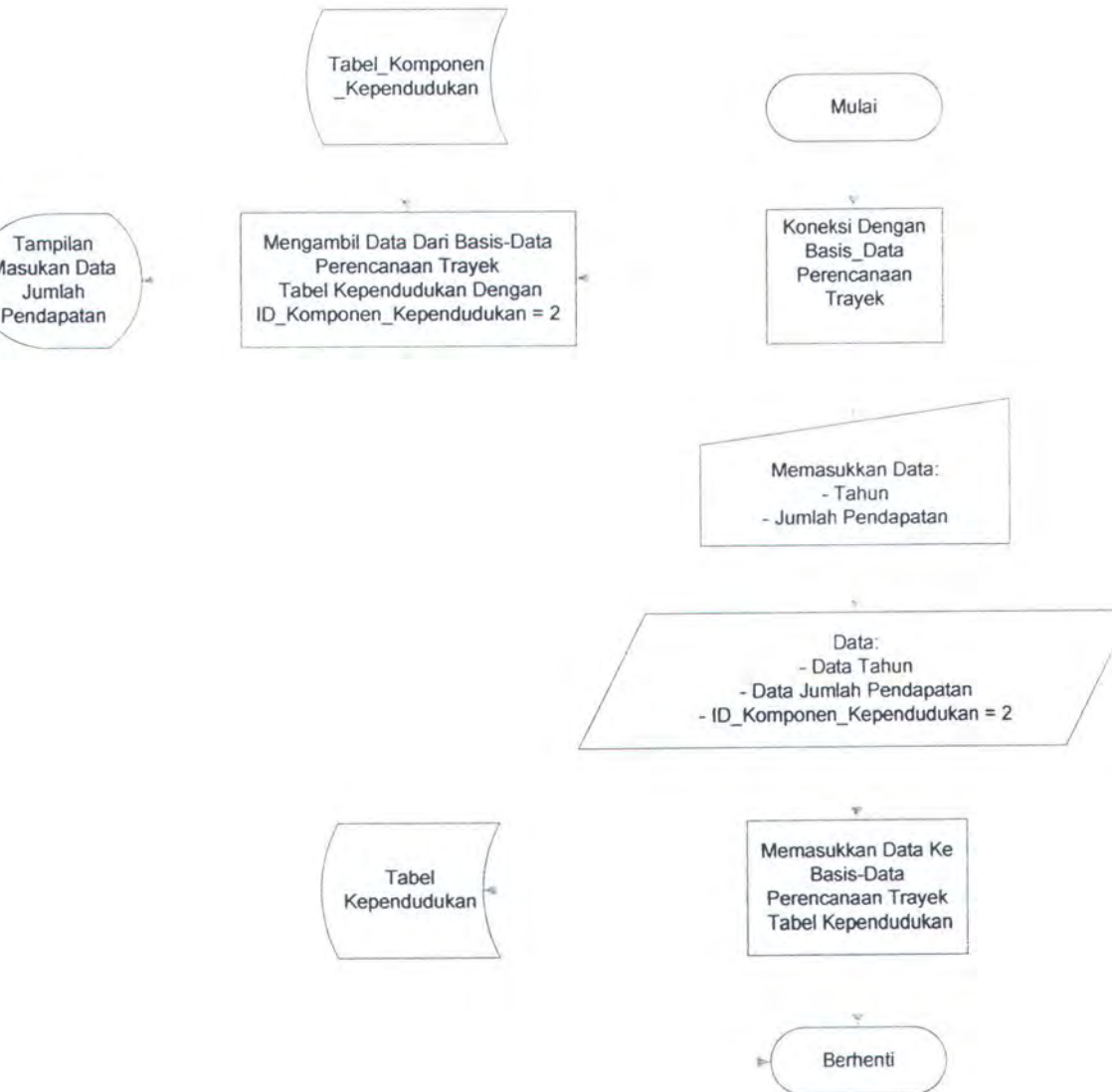


Gambar 3.25 Flowchart Proses Masukan Data Jumlah Penduduk (Proses 1.1.1)

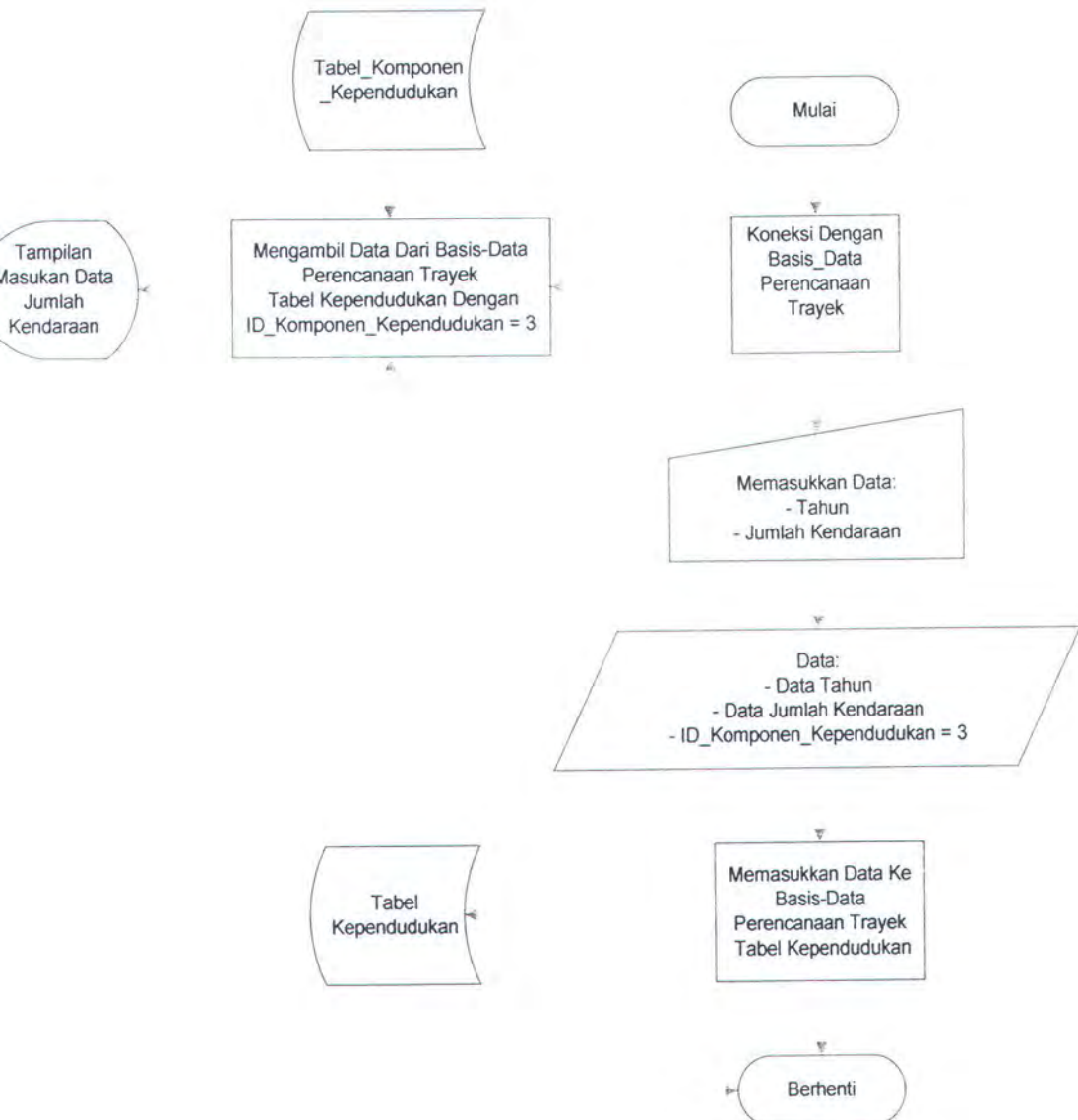
3. 2. 1. 1. 2. Proses Masukan Data Jumlah Pendapatan

Proses ini adalah proses untuk memasukkan data jumlah pendapatan untuk

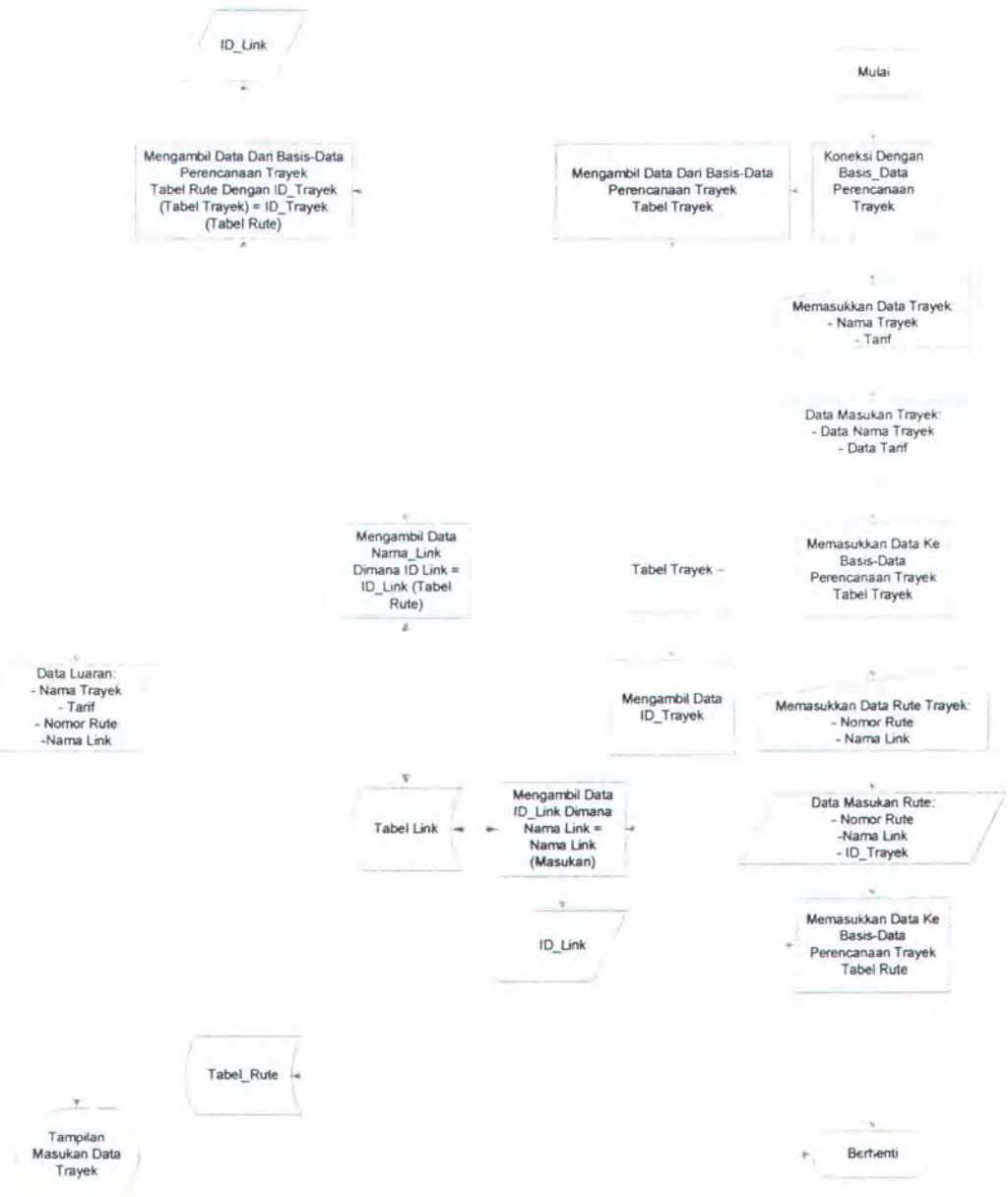
ke dalam database. Hasil dari proses ini nantinya akan dimasukkan ke dalam



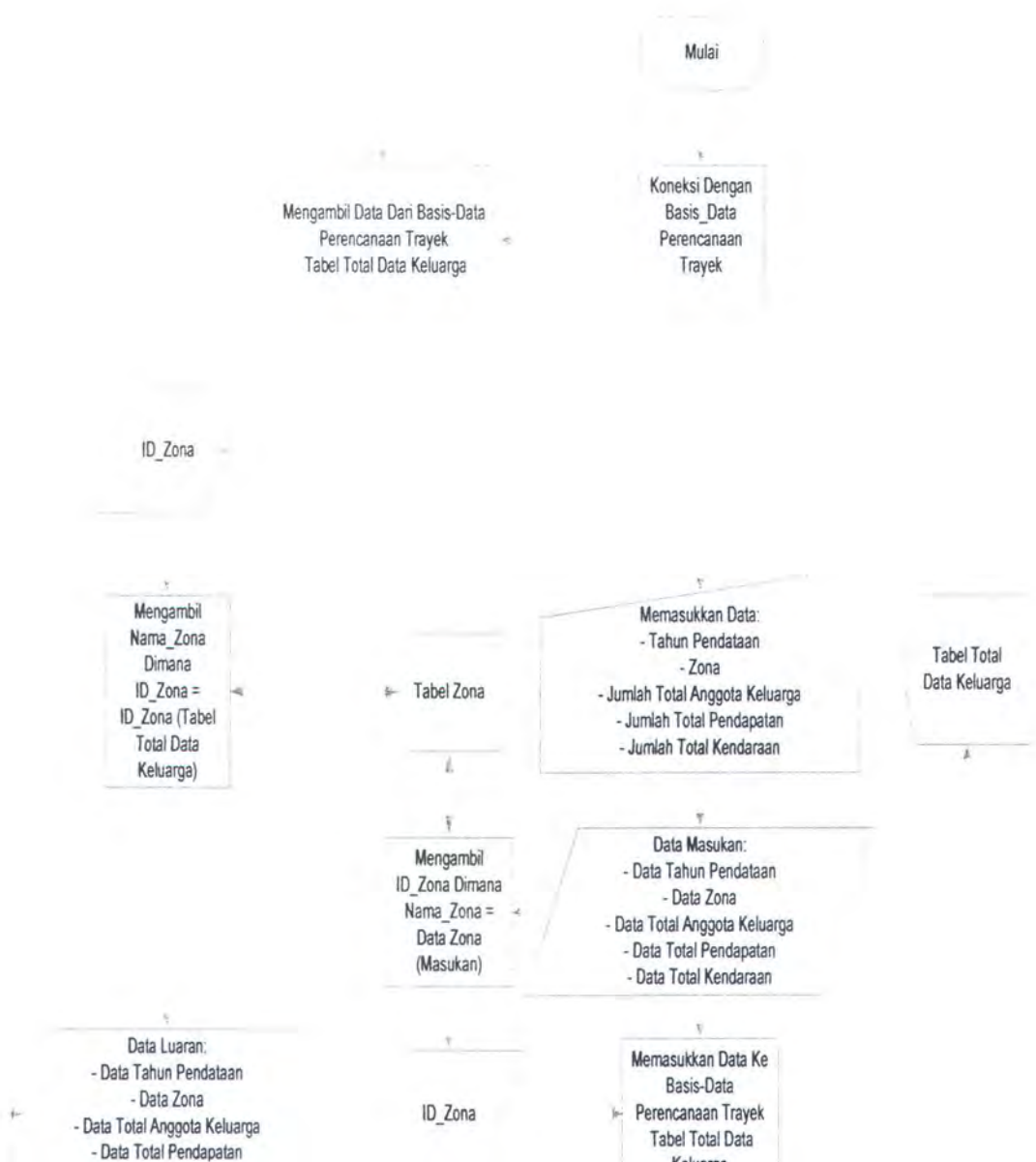
Gambar 3.26 Flowchart Proses Masukan Data Pendapatan (Proses 1.1.2)

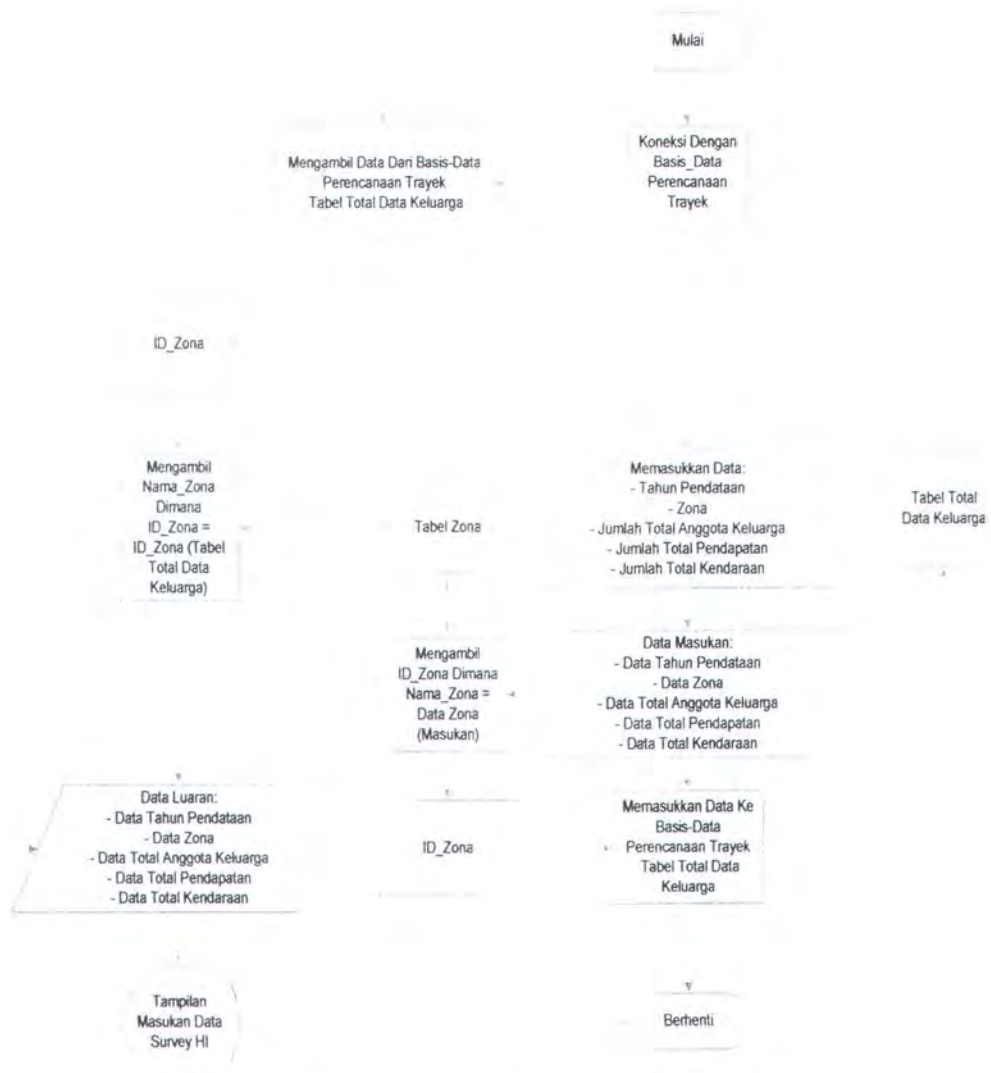


Gambar 3.27 Flowchart Proses Masukan Data Kendaraan (Proses 1.1.3)



Gambar 3.28 Flowchart Proses Masukan Data Trayek (Proses 1.2)

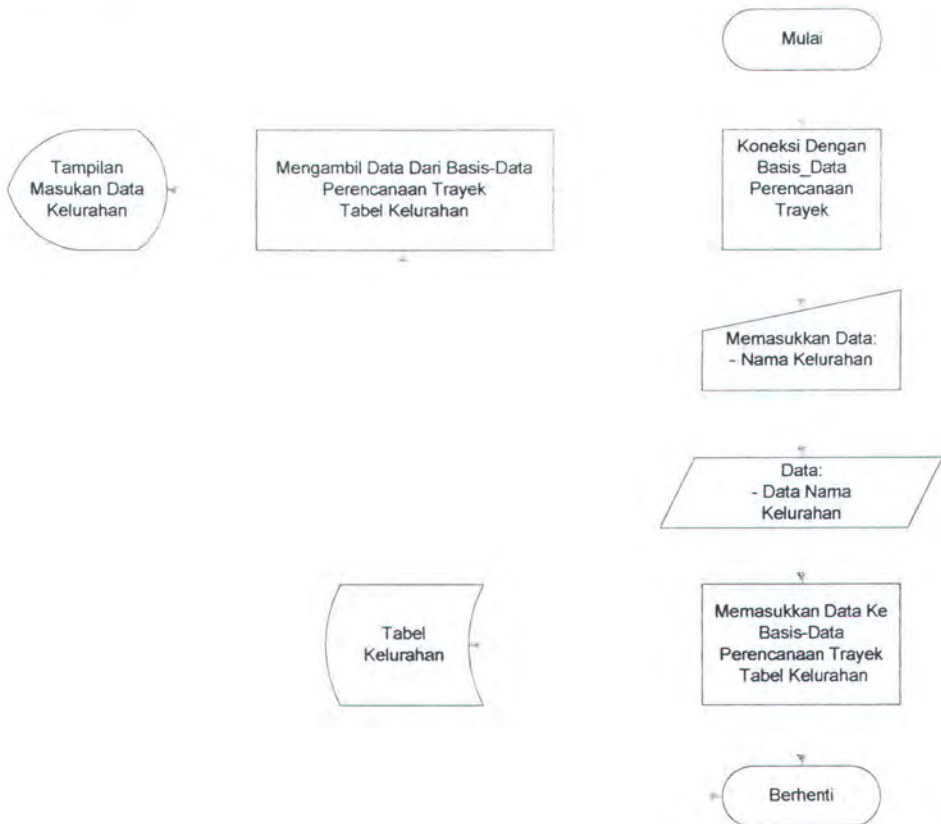




Gambar 3.31 Flowchart Proses Masukan Asal Tujuan Perjalanan (Proses 1.3.2)

3. 2. 1. 4. 3. Masukan Data Kelurahan

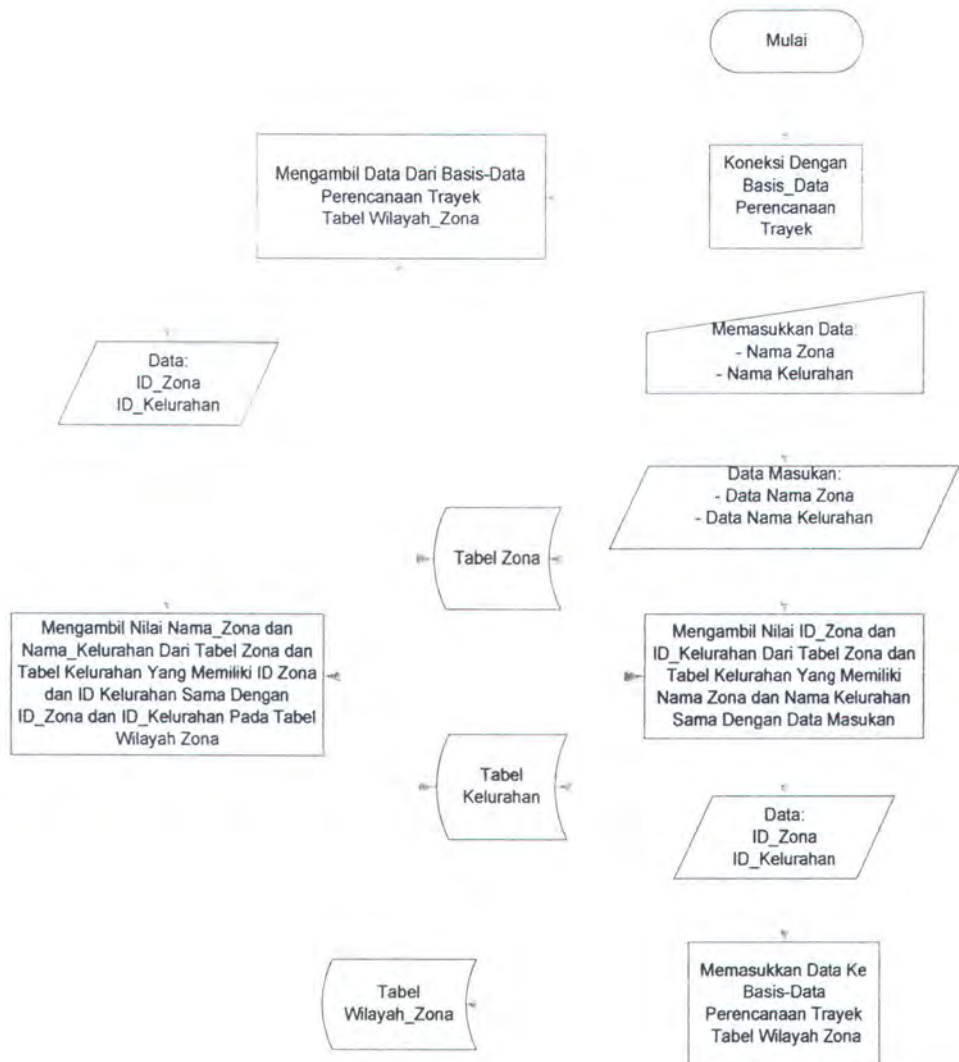
Proses masukan data kelurahan merupakan proses masukan data kelurahan yang ada di Kota Mojokerto ke dalam tabel kelurahan. Data yang dimasukkan tersebut adalah data nama kelurahan. Proses detailnya dapat dilihat pada *flowchart* gambar 3.36.



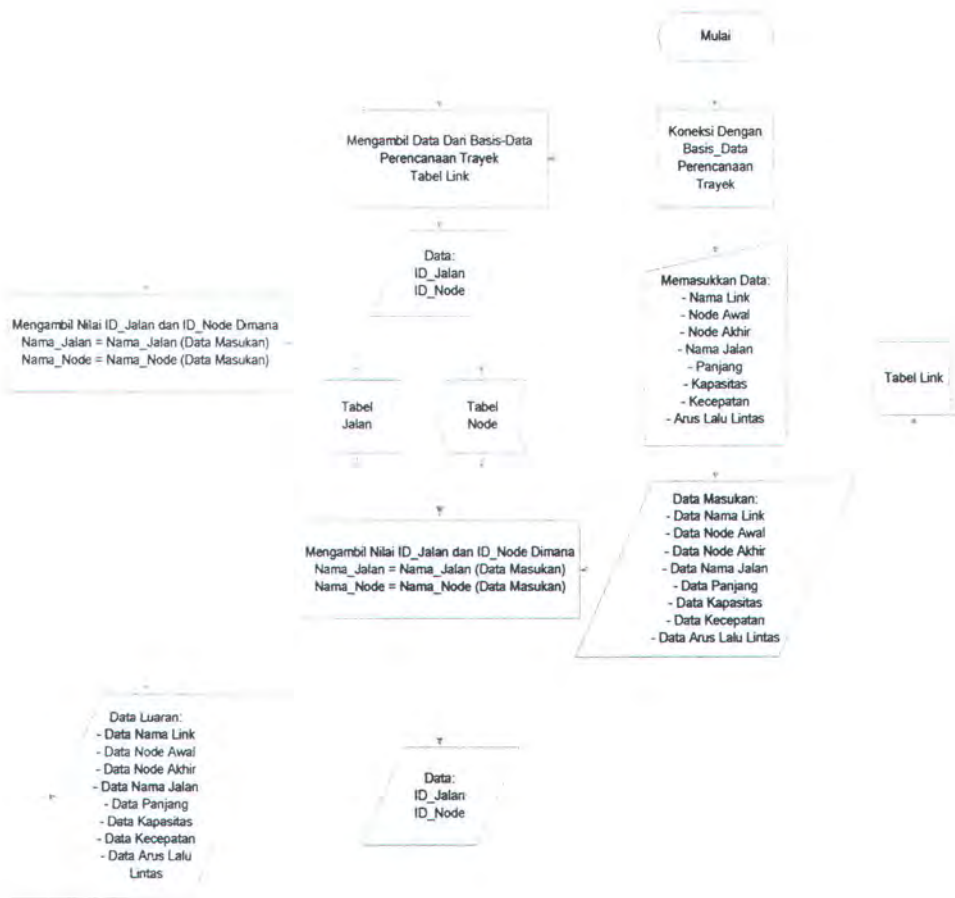
Gambar 3.36 *Flowchart* Proses Data Kelurahan (Proses 1.4.3)

3. 2. 1. 4. 4. Masukan Data Wilayah Zona

hir, nama jalan, panjang link, kapasitas link, kecepatan kendaraan di link tersebut, dan arus lalu lintasnya. Proses detailnya dapat dilihat pada *flowchart* gambar 3.38.



ma zona tempat node tersebut, dan apakah node tersebut termasuk node pusat atau bukan. Proses detailnya dapat dilihat pada *flowchart* gambar 3.39.





Gambar 3.39 Flowchart Proses Masukan Data Node (Proses 1.4.6)

3. 2. 2. Proses Perencanaan Trayek

Yang dilakukan dalam proses perencanaan trayek adalah mendapatkan rute baru. Dimana dalam mendapatkan atau menentukan rute baru trayek angkutan kota terdapat beberapa hal yang perlu diperhatikan, yaitu:

Jarak dan waktu merupakan salah satu faktor penentu rute itu nantinya akan diminati atau tidak, semakin kecil rasio waktu perjalanan, yang merupakan perbandingan antara waktu perjalanan angkutan kota dengan waktu perjalanan

Rute baru tidak boleh berbeda jauh dengan rute yang lama, karena pada dasarnya pada rute lama tersebut banyak terdapat penumpang, karena itu rute lama harus dijadikan rute dasar dalam penentuan rute baru trayek angkutan kota.

Kemudian langkah-langkah yang dilakukan dalam penentuan rute angkutan kota secara garis besar adalah:

Menentukan rute tercepat antara zona satu dengan zona lain. Sehingga akan didapatkan rute-rute alternatif yang kemudian digabungkan. Dalam mencari rute tercepat tersebut digunakan algoritma dijkstra, yaitu salah satu algoritma yang digunakan untuk mendapatkan rute terpendek

Mencari himpitan yang terjadi antara rute-rute alternatif yang ada dengan trayek lama.

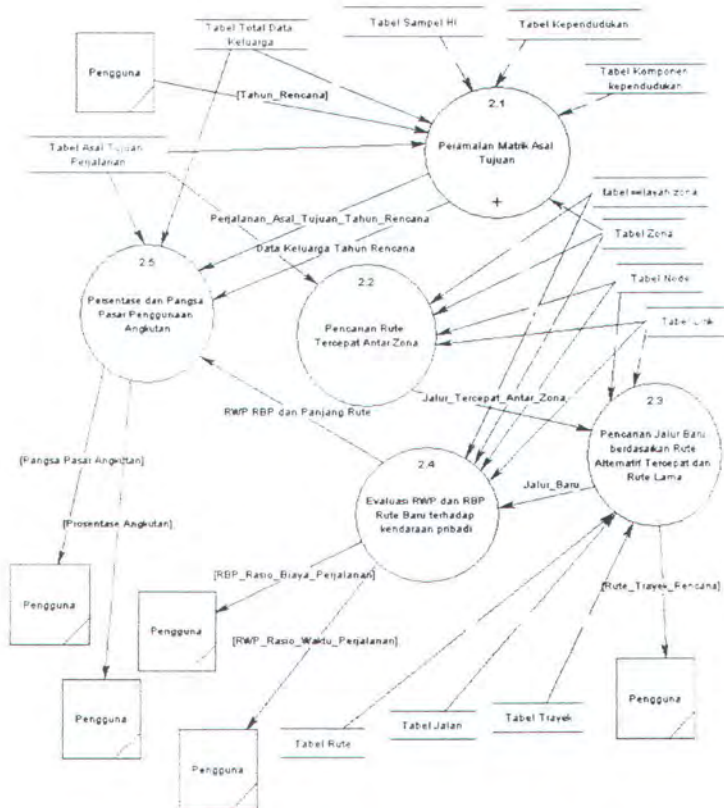
Bila antara rute lama dan rute alternatif tidak ada himpitan, maka dicari rute terpendek antara titik yang tidak berhimpitan tersebut.

Didapatkan rute terpendek untuk satu trayek.

Lakukan langkah nomor b sampai d hingga semua trayek yang ada dicari rute barunya.

Kemudian lakukan evaluasi trayek baru dengan mencari RWP (Rasio Waktu Perjalanan) dan RBP (Rasio Biaya Perjalanan) untuk perjalanan tiap trayek angkutan kota pada tiap perpotongan perjalanan antara 2 pusat zona yang dilalui trayek tersebut.

Dengan diketahuinya jumlah perjalanan total yang didapat dari matrik asal tujuan perjalanan antar zona dan RWP serta RBP, maka dicari persentase

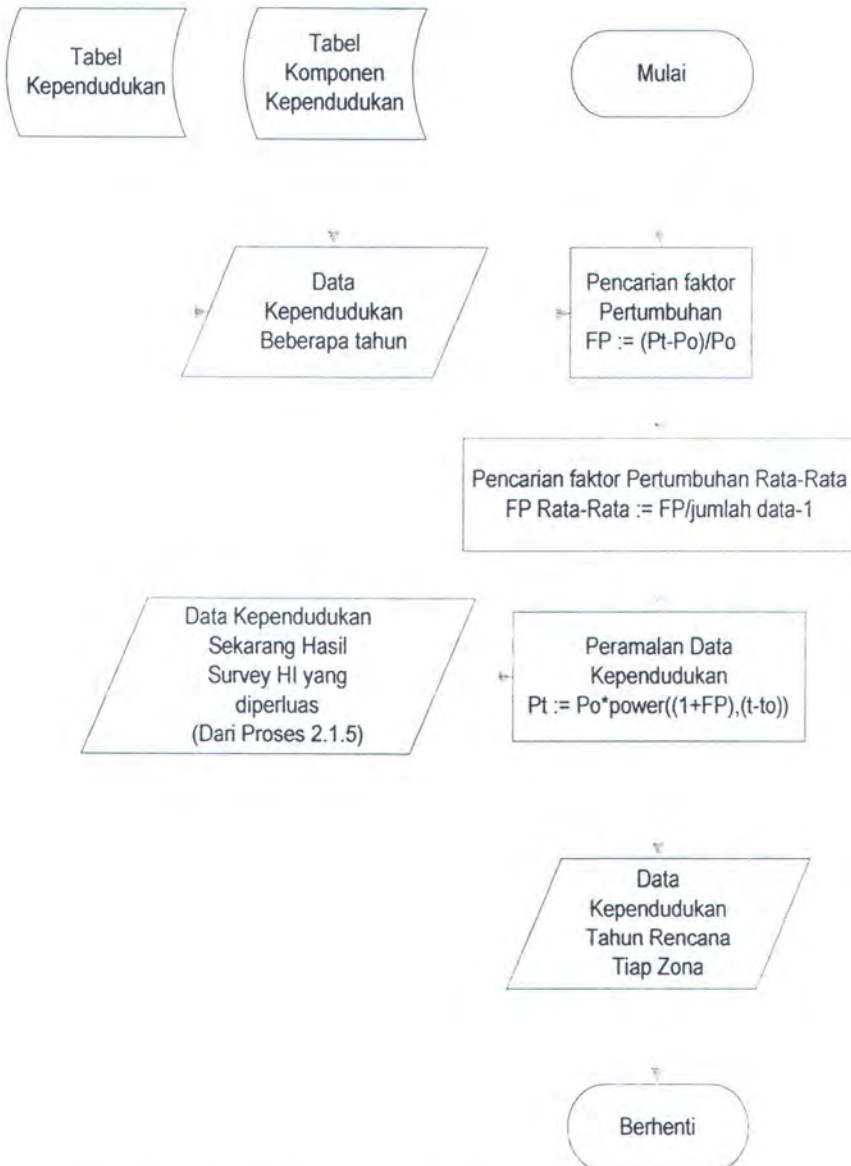


Gambar 3.41 DFD Level 2 Perencanaan Trayek Angkutan Kota

3. 2. 2. 1. Peramalan Matrik Asal Tujuan Perjalanan

Proses peramalan matrik asal tujuan perjalanan merupakan DFD level 3 yang mempunyai proses sebagai berikut:

- Proses 2.1.1. : Peramalan Data Kependudukan.
- Proses 2.1.2. : Pencarian Persamaan Regresi.
- Proses 2.1.3. : Peramalan Bangkitan Perjalanan.



Gambar 3.43 Flowchart Proses Peramalan Data Kependudukan (Proses 2.1.1)

Tabel Total
Data Keluarga

mulai

Data
Keluarga

Korelasi untuk variabel bebas dan tidak bebas

$$r = \frac{N \cdot \sum_i (X_i Y_i) - \sum_i (X_i) \sum_i (Y_i)}{\left[\left\{ N \sum_i (X_i^2) - \left(\sum_i (X_i) \right)^2 \right\} \left\{ N \sum_i (Y_i^2) - \left(\sum_i (Y_i) \right)^2 \right\} \right]^{\frac{1}{2}}}$$

Data Asal
Tujuan
Perjalanan

Tabel Asal
Tujuan
Perjalanan

Uji Korelasi t-test

$$t_{\alpha} = \frac{r \sqrt{n-2}}{\sqrt{1-r^2}}$$

$R_{xy} > 0$ dan $|t_{\alpha}| > t(0.025, n-2)$

Ya

Variabel
yang
diterima

Mengurutkan variabel bebas yang
diterima berdasarkan korelasi
dengan variabel tidak bebas dari
besar ke kecil

$i := 1$
 $j := i+1$

$R_{xy}[i] < R_{xy}[j]$ Tidak

Ya

$a := R_{xy}[i]$
 $R_{xy}[i] := R_{xy}[j]$
 $R_{xy}[j] := a$

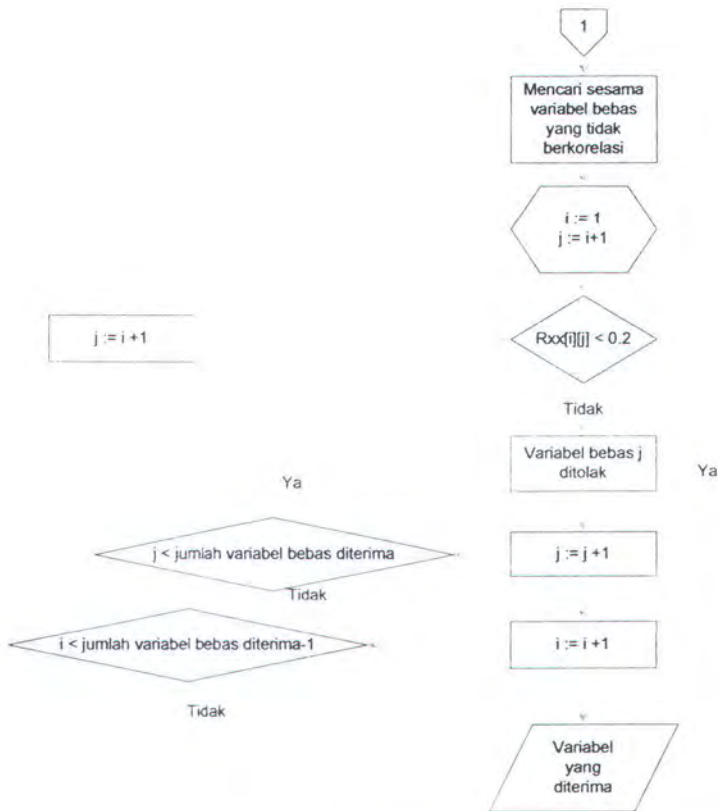
$j := j + 1$

Ya

$j < \text{jumlah variabel bebas diterima}$

Tidak

$j := i + 1$

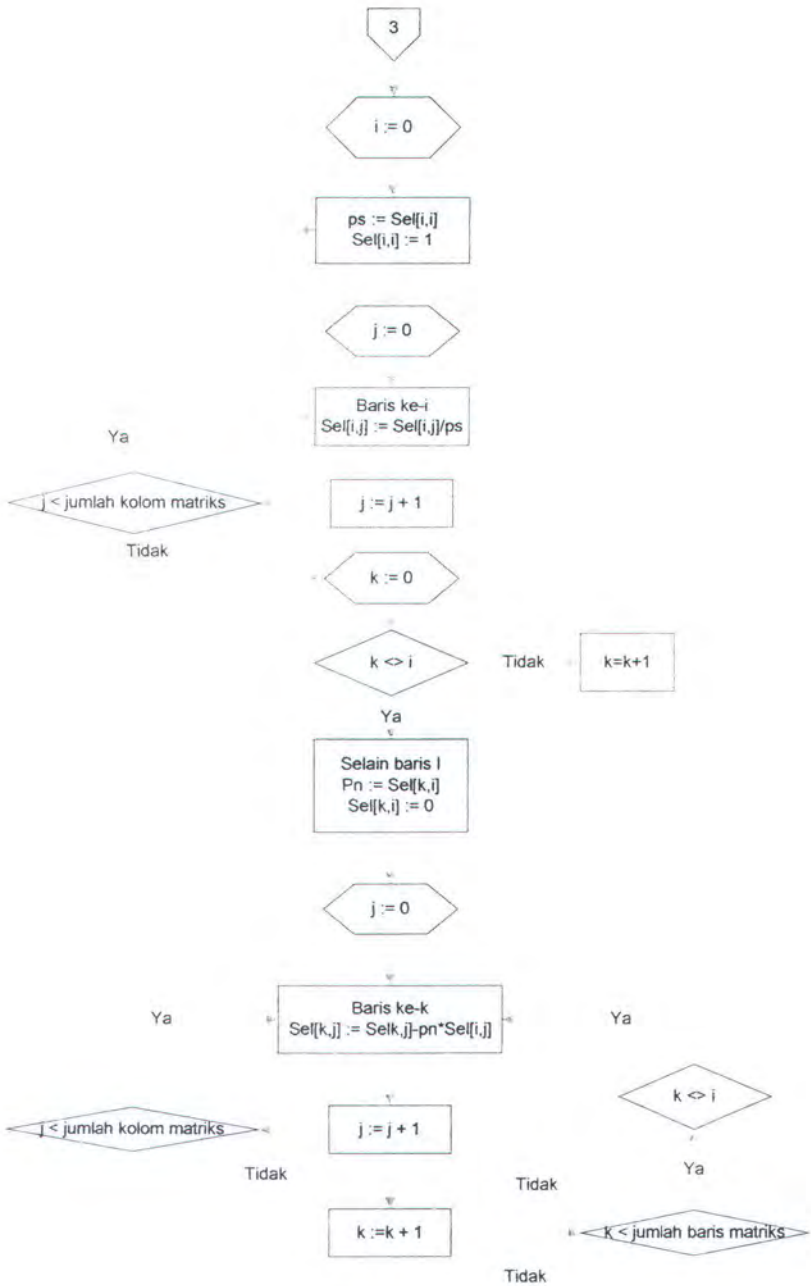


Membuat Matriks A dengan x adalah variabel bebas yang diterima

$$A = X'X = \begin{bmatrix} n & \sum_{i=1}^n X_{1i} & \sum_{i=1}^n X_{2i} & \cdots & \sum_{i=1}^n X_{ki} \\ \sum_{i=1}^n X_{1i} & \sum_{i=1}^n X_{1i}^2 & \sum_{i=1}^n X_{1i}X_{2i} & \cdots & \sum_{i=1}^n X_{1i}X_{ki} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n X_{ki} & \sum_{i=1}^n X_{ki}X_{1i} & \sum_{i=1}^n X_{ki}X_{2i} & \cdots & \sum_{i=1}^n X_{ki}^2 \end{bmatrix}$$

Matriks g

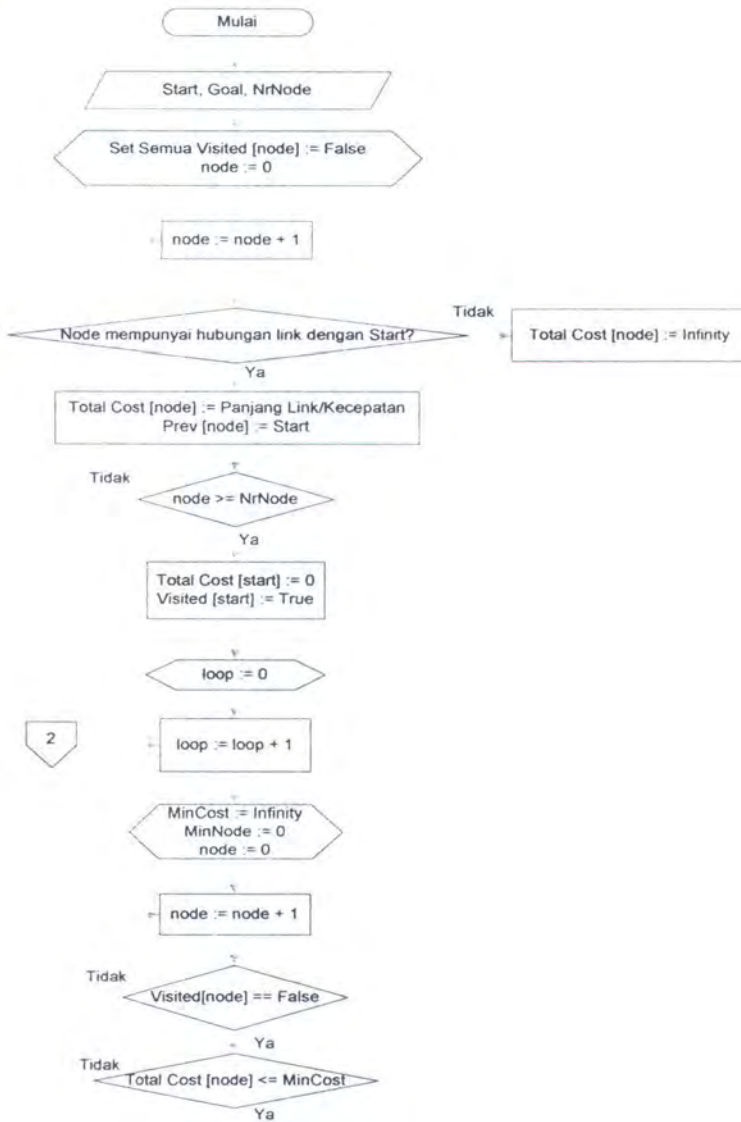
$$g = X'y = \begin{bmatrix} g_0 = \sum_{i=1}^n y_i \\ g_1 = \sum_{i=1}^n x_{1i}y_i \\ \vdots \\ g_k = \sum_{i=1}^n x_{ki}y_i \end{bmatrix}$$

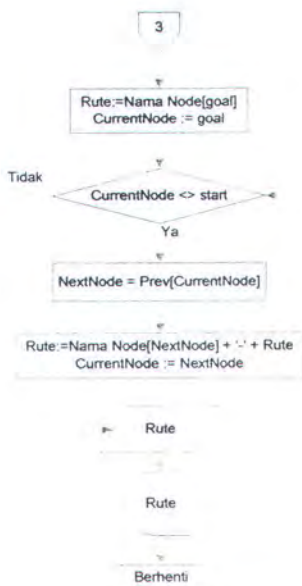


ng didapatkan pada proses 2.1.3 sebelumnya. Nilai peramalan bangkitan dan ai peramalan tarikan tiap zona yang digunakan sebagai pembanding dianggap na, dalam hal ini diasumsikan perjalanan yang terjadi adalah perjalanan *Home use Trip* atau perjalanan berbasis rumah, dimana semua perjalanan yang terjadi anggap dimulai dari rumah dan akan kembali ke rumah lagi. Untuk proses lebih asnya dapat dilihat pada *flowchart* gambar 3.49.

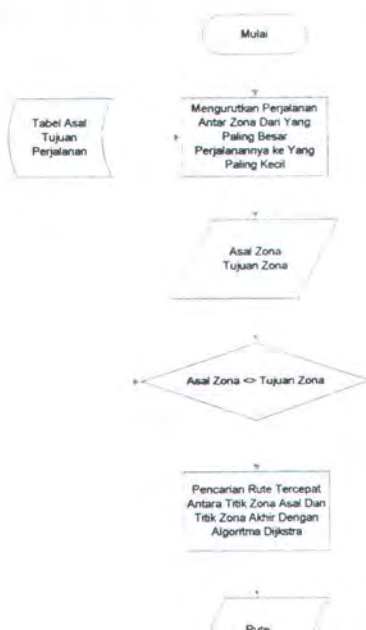


menggambarkan algoritma tersebut dapat dilihat pada gambar 3.51 sampai 3.53. Sedangkan flowchart untuk proses pencarian rute tercepat dapat dilihat pada gambar 3.54.





Gambar 3.53 Flowchart Algoritma Dijkstra (3)



3.2.2.3. Pencarian Jalur Baru Berdasarkan Rute Alternatif Tercepat dan Rute Lama

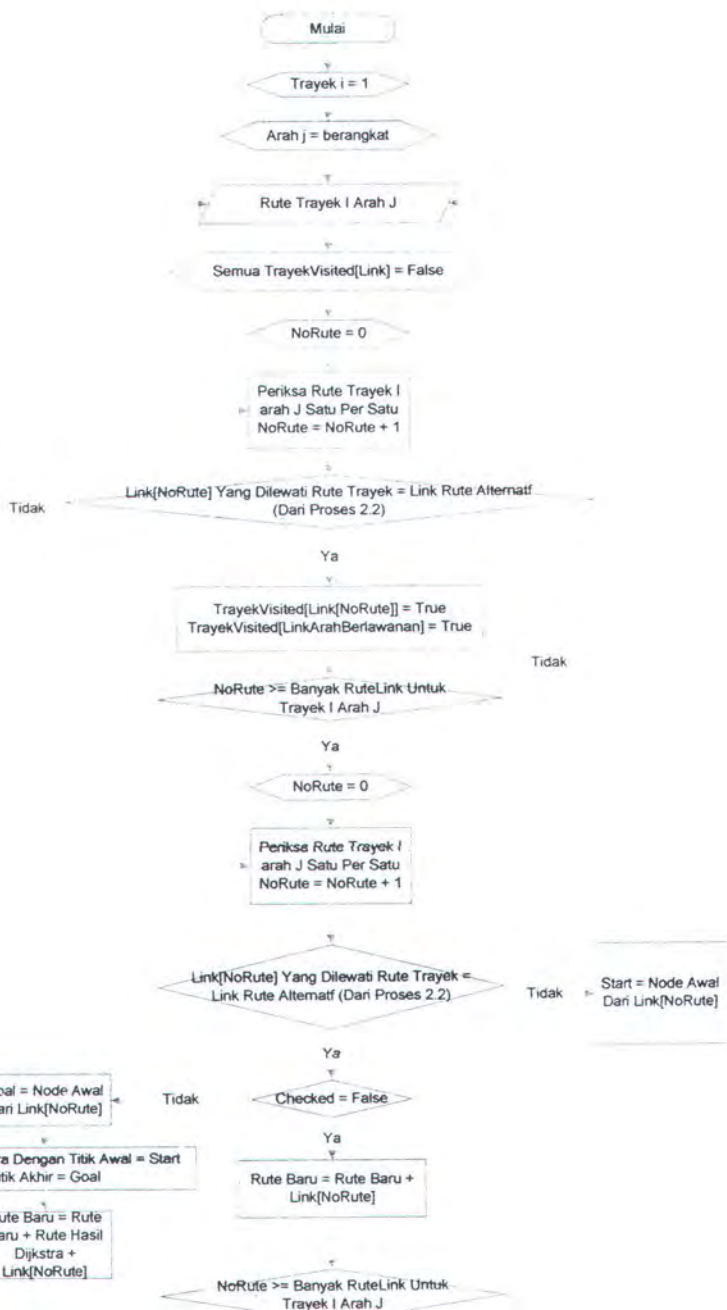
Pada proses ini dilakukan pencarian jalur baru trayek angkutan kota yang berdasarkan pada rute lama trayek angkutan kota dan alternatif rute tercepat antara zona. Dimana jalur yang dilewati rute lama dan alternatif rute tercepat antara zona bersamaan dijadikan jalur dari rute baru yang sedang dibentuk. Untuk jalur yang tidak dilewati rute lama dan rute alternatif tercepat antara zona, maka pada titik antar perpotongan yang terjadi dicari rute tercepatnya berdasarkan waktu perjalanan dengan menggunakan algoritma *dijkstra*. Pada dasarnya algoritma *dijkstra* yang digunakan sama dengan algoritma *dijkstra* yang telah dijelaskan sebelumnya, hanya pada algoritma *dijkstra* untuk mencari rute tercepat antara dua titik perpotongan terdapat penambahan syarat, yaitu link yang boleh dimasukkan ke dalam rute adalah link yang belum pernah dilewati oleh rute trayek tersebut. Maka untuk lebih jelasnya algoritma *dijkstra* untuk mencari rute tercepat antara dua titik perpotongan tersebut dapat dilihat pada gambar 3.55 sampai 3.57. Sedangkan proses pencarian jalur barunya dapat dilihat pada *flowchart* gambar 3.58

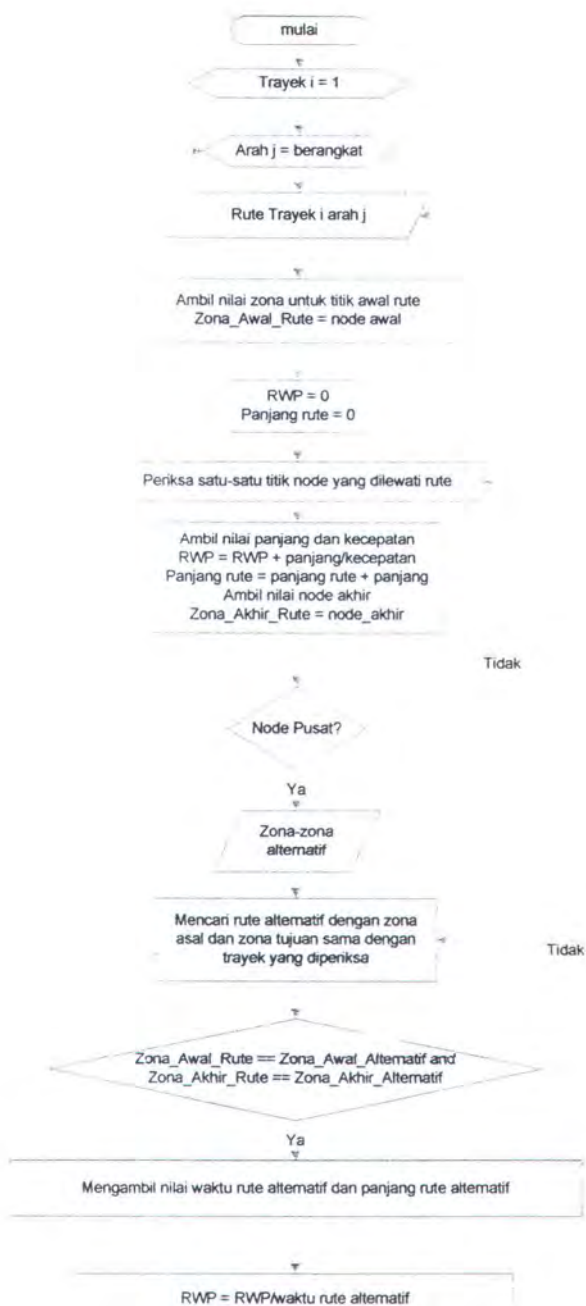
3.2.2.4. Evaluasi RWP dan RBP Rute Baru Terhadap Kendaraan Bermotor

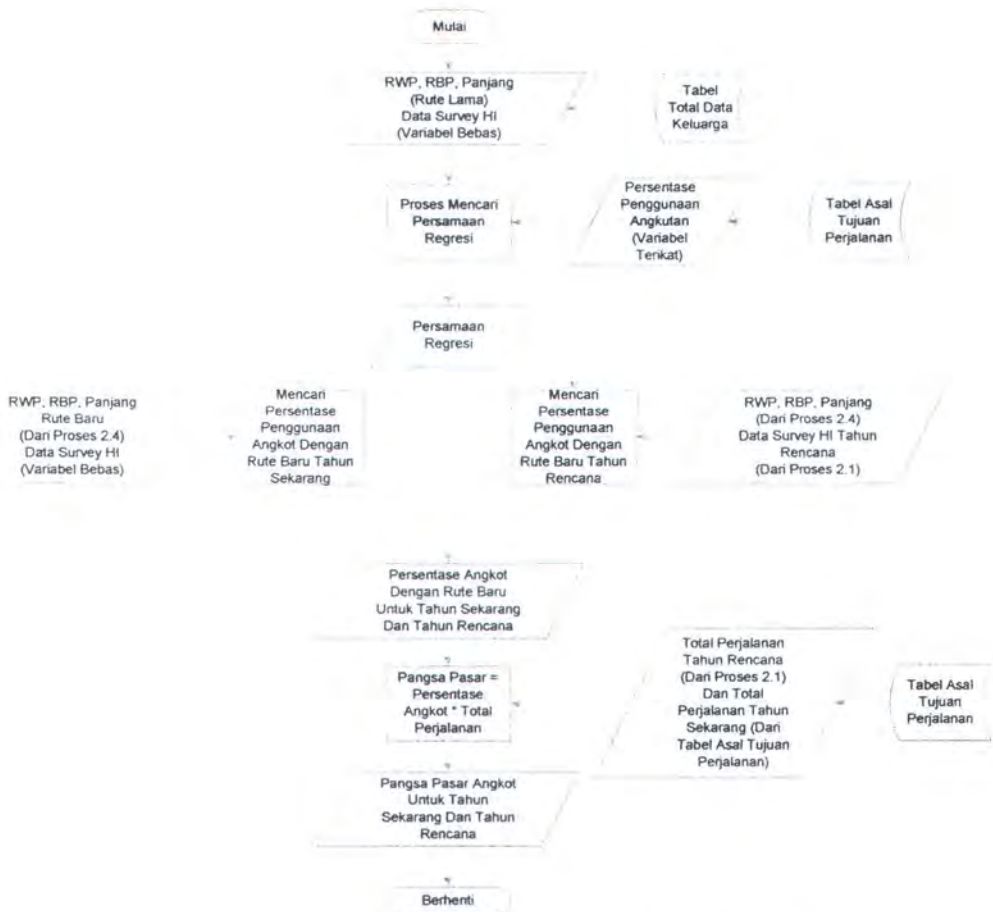
Pada proses ini dilakukan evaluasi RWP (Rasio Waktu Perjalanan) dan RBP (Rasio Biaya Perjalanan) untuk tiap trayek. Perhitungan RWP dan RBP ini dilakukan pada jalur antara zona yang dilewati trayek tersebut. Misalnya suatu trayek A melewati zona 1 – zona 2 – zona 3. Maka RWP dan RBP dihitung untuk perjalanan dari zona 1 ke zona 2 dan zona 2 ke zona 3. proses lebih jelasnya

ngurangi jumlah penumpang (memperendah prosentase penggunaan angkutan umum). Prosentase ini dicari dengan menggunakan persamaan regresi linier ganda. Yaitu dengan meregresikan faktor data kependudukan seperti pendapatan, kendaraan, penduduk dengan RWP, RBP, panjang perjalanan yang jadi dan prosentase penggunaan angkutan umum. Proses lebih jelasnya dapat dilihat pada gambat 3.60.









Gambar 3.60 Flowchart Proses Prosentase dan Pangsa Pasar Penggunaan Angkutan Kota (Proses 2.5)

4. PERANCANGAN ANTAR MUKA

Pada sub-bab ini membahas perancangan antarmuka untuk aplikasi perencanaan trayek. Terdapat 2 jenis antarmuka dalam aplikasi untuk perencanaan

Aplikasi untuk perencanaan trayek adalah halaman utama yang akan menampilkan menu yang ada pada aplikasi untuk perencanaan trayek angkutan kota, dengan *layout* seperti pada gambar 3.62. Menu-menu yang ada pada halaman utama tersebut adalah menu masukan dan menu proses perencanaan ulang trayek angkutan kota. Menu masukan mempunyai sub-menu didalamnya yang akan membawa pengguna ke dalam halaman dari sub-menu yang dipilih. Sub-menu dari menu masukan tersebut adalah:

Masukan Data Kependudukan, yang terdiri dari masukan jumlah penduduk, masukan jumlah pendapatan, dan masukan jumlah kendaraan.

Masukan Data Trayek.

Masukan Data Survey HI, yang terdiri dari dari masukan survey HI, masukan asal tujuan perjalanan dan masukan sampel survey HI.

Dan yang terakhir masukan data wilayah, yang terdiri dari masukan data jalan, masukan data kelurahan, masukan data zona, masukan data wilayah zona, masukan data *link* dan masukan data *node*.

Dimana pada dasarnya *layout* untuk antarmuka menu masukan tersebut mempunyai pola yang sama, seperti dapat dilihat pada gambar 3.63.

Kemudian pada menu proses perencanaan ulang trayek angkutan kota dapat antarmuka masukan tahun untuk proses perencanaan ulang trayek angkutan kota, dengan *layout* seperti pada gambar 3.64, dan antarmuka hasil Moran proses perencanaan ulang trayek angkutan kota, dengan *layout* seperti pada gambar 3.65.

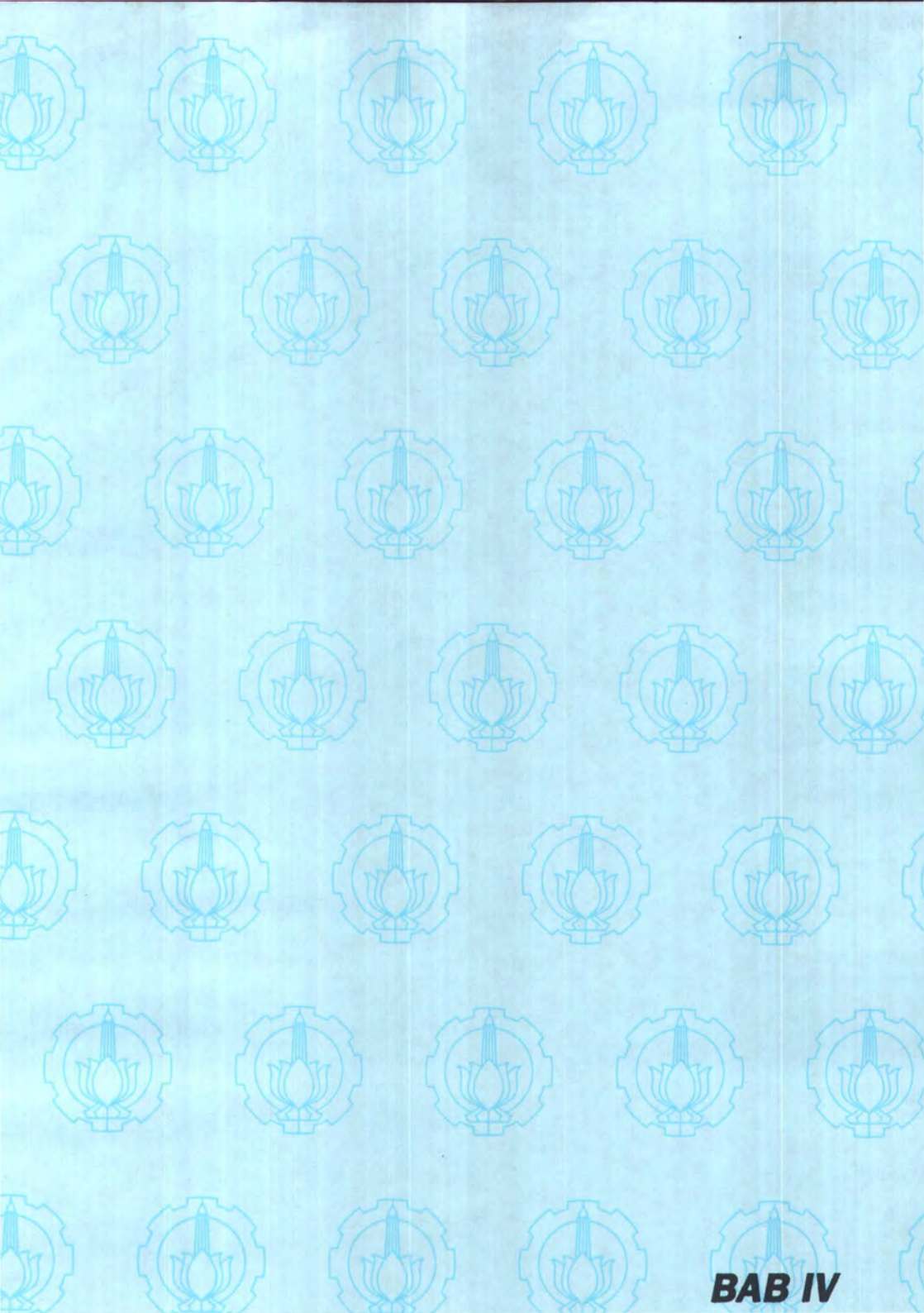
JUDUL
MASUKAN DATA TABEL
TOMBOL PENGATUR
MENU

Gambar 3.63 *Layout* Antarmuka Menu Masukan

JUDUL
MASUKAN DATA PROSES
TOMBOL PROSES
MENU

Gambar 3.64 *Layout* Antarmuka Masukan Data Proses

JUDUL



BAB IV

PEMBUATAN APLIKASI

Pada bab ini dijelaskan tentang pembuatan aplikasi yang terdiri dari tiga bagian, yaitu pembuatan basis-data, pembuatan proses, dan pembuatan antarmuka. Dalam pembuatan basis-data digunakan Microsoft SQL Server sedangkan untuk pembuatan proses serta antarmukanya digunakan Borland Delphi 7.

4.1. PEMBUATAN BASIS DATA

Dalam pembuatan basis-data, yang digunakan sebagai tempat penyimpanan data-data untuk proses aplikasi yang akan dibuat, digunakan Microsoft SQL Server. Pada basis-data ini terdapat 13 tabel, yaitu tabel zona, tabel *node*, tabel jalan, tabel *link*, tabel kelurahan, tabel wilayah zona, tabel keluarga, tabel asal tujuan perjalanan, tabel sampel HI, tabel trayek, tabel rute, tabel kependudukan, dan tabel komponen kependudukan. Selanjutnya tabel-tabel ini dimasukkan ke dalam sebuah basis-data 'PERENCANAAN TRAYEK' dengan menggunakan perintah sql yang dapat dilihat pada program 4.1 sampai dengan program 4.13.

```
1 CREATE TABLE [dbo].[Zona] (  
2 [id_zona] [int] IDENTITY (1, 1) NOT FOR REPLICATION NOT NULL ,  
3 [nama_zona] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS  
4 ) ON [PRIMARY]  
5 GO  
6 ALTER TABLE [dbo].[Zona] WITH NOCHECK ADD  
7 CONSTRAINT [PK_zona] PRIMARY KEY CLUSTERED  
8 (  
9 [id_zona]  
10 ) ON [PRIMARY]  
11 GO
```


dengan nilai awal satu dan kenaikannya sebesar satu nilai juga (baris 2 program 4.1). *Field* nama_zona bertipe *varchar* dengan panjang 50, dan diperbolehkan kosong (baris 3 program 4.1). Terdapat *constraint* untuk penambahan *field* id_zona sebagai *Primary Key* tabel zona (baris 7 sampai baris 10 program 4.1).

```

1 CREATE TABLE [dbo].[Kelurahan] (
2   [id_kelurahan] [int] IDENTITY (1, 1) NOT FOR REPLICATION NOT NULL
3   ,
4   [nama_kelurahan] [varchar] (50) COLLATE
5   SQL_Latin1_General_CP1_CI_AS NULL
6   ) ON [PRIMARY]
7 GO
8 ALTER TABLE [dbo].[Kelurahan] WITH NOCHECK ADD
9 CONSTRAINT [PK_Kelurahan] PRIMARY KEY CLUSTERED
10 (
11  [id_kelurahan]
12 ) ON [PRIMARY]
13 GO

```

Program 4.2. Perintah SQL untuk membuat tabel kelurahan beserta *constraint*-nya

Seperti halnya pada program 4.1 untuk membuat tabel zona, pada program 4.2. merupakan bahasa sql untuk membuat tabel kelurahan, dengan id_kelurahan dan nama_kelurahan (baris 2 dan baris 3 program 4.2.) sebagai *field-field*-nya.

```

1 CREATE TABLE [dbo].[Wilayah_Zona] (
2   [id_zona] [int] NOT NULL ,
3   [id_kelurahan] [int] NOT NULL
4   ) ON [PRIMARY]
5 GO
6 ALTER TABLE [dbo].[Wilayah_Zona] ADD
7 CONSTRAINT [FK_Wilayah_Zona_Kelurahan] FOREIGN KEY
8 ([id_kelurahan]
9 ) REFERENCES [dbo].[Kelurahan] (
10  [id_kelurahan]
11 ) ON DELETE CASCADE ON UPDATE CASCADE ,
12 CONSTRAINT [FK_Wilayah_Zona_Zona] FOREIGN KEY
13 ([id_zona]
14 ) REFERENCES [dbo].[Zona] (
15  [id_zona]

```

(baris 2 dan baris 3 program 4.3). Selain itu terdapat pula dua buah *constraint*, yaitu FK Wilayah Zona Kelurahan dan FK Wilayah Zona Zona. FK Wilayah Zona Kelurahan menambahkan *field* id kelurahan sebagai kunci tamu (*Foreign Key*) ke dalam tabel wilayah_zona dan mengacu pada tabel kelurahan dengan *field* ID_Kelurahan. *Constraint* ini dalam ID_Kelurahan kondisi ON DELETE CASCADE dan ON UPDATE CASCADE, dimana bila pada tabel kelurahan dihapus atau diganti maka ID_Kelurahan pada tabel wilayah zona juga akan ikut terhapus atau terganti. Begitu pula FK Wilayah Zona Zona mempunyai fungsi yang hampir sama dengan FK Wilayah Zona Kelurahan, hanya FK Wilayah Zona Zona mengacu pada tabel zona (baris 7 sampai baris 16 program 4.3).

```

1 CREATE TABLE [dbo].[Node] (
2   [ID_Node] [int] NOT NULL ,
3   [Nama_Node] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
   NULL ,
4   [ID_Zona] [int] NULL ,
5   [Node_Pusat] [bit] NULL
6 ) ON [PRIMARY]
7 GO
8 ALTER TABLE [dbo].[Node] WITH NOCHECK ADD
9   CONSTRAINT [PK_Node] PRIMARY KEY CLUSTERED
10  (
11    [ID_Node]
12  ) ON [PRIMARY]
13 GO
14 ALTER TABLE [dbo].[Node] ADD
15   CONSTRAINT [FK_Node_Zona] FOREIGN KEY
16  (
17    [ID_Zona]
18  ) REFERENCES [dbo].[Zona] (
19    [id_zona]
20  ) ON DELETE CASCADE ON UPDATE CASCADE
21 GO

```

Program 4.4 Perintah SQL untuk membuat tabel *node* beserta *constraint*nya

```

1  CREATE TABLE [dbo].[Total_Data_Keluarga] (
2  [ID_Total_Data_Keluarga] [int] IDENTITY (1, 1) NOT FOR
  REPLICATION NOT NULL ,
3  [ID_Zona] [int] NULL ,
4  [Tahun_Pendataan] [int] NULL ,
5  [Total_Anggota_Keluarga] [float] NULL ,
6  [Total_Pendapatan] [float] NULL ,
7  [Total_Kendaraan] [float] NULL
8  ) ON [PRIMARY]
9  GO
10 ALTER TABLE [dbo].[Total_Data_Keluarga] WITH NOCHECK ADD
11 CONSTRAINT [PK_Total_Data_Keluarga] PRIMARY KEY CLUSTERED
12 (
13 [ID_Total_Data_Keluarga]
14 ) ON [PRIMARY]
15 GO
16 ALTER TABLE [dbo].[Total_Data_Keluarga] ADD
17 CONSTRAINT [FK_Total_Data_Keluarga_Zona] FOREIGN KEY
18 (
19 [ID_Zona]
20 ) REFERENCES [dbo].[Zona] (
21 [id_zona]
22 ) ON DELETE CASCADE ON UPDATE CASCADE
23 GO

```

Program 4.7 Perintah SQL untuk membuat tabel keluarga beserta *constraint*-nya

Program 4.7 merupakan program sql untuk membuat tabel Total_Data_Keluarga, dengan *field-field*-nya ID_Total_Data_Keluarga, ID_Zona, Tahun_Pendataan, Total_Anggota_Keluarga, Total_Kendaraan, dan Total_Pendapatan. Dan mempunyai dua buah *constraint* PK_Result dan FK_Total_Data_Keluarga_Zona.

Program 4.8 merupakan program sql untuk membuat tabel link dengan *field-field*-nya ID_Link, Nama_Link, Panjang, Arus_Lalu_Lintas, Kapasitas, Kecepatan, ID_Node_Awal, ID_Node_Akhir, dan ID_Jalan. Serta terdiri pula dari 4 *constraint* yaitu PK_Link, FK_Link_Jalan yang mengacu pada tabel jalan, FK_Link_Node dan FK_Link_Nodel yang mengacu pada tabel node.


```

1  CREATE TABLE [dbo].[Komponen_Kependudukan] (
2  [ID_Komponen_Kependudukan] [int] IDENTITY (1, 1) NOT FOR
   REPLICATION NOT NULL ,
3  [Nama_Komponen_Kependudukan] [varchar] (50) COLLATE
   SQL_Latin1_General_CP1_CI_AS NULL
4  ) ON [PRIMARY]
5  GO
6  ALTER TABLE [dbo].[Komponen_Kependudukan] WITH NOCHECK ADD
7  CONSTRAINT [PK_Komponen_Kependudukan] PRIMARY KEY CLUSTERED
8  (
9  [ID_Komponen_Kependudukan]
10 ) ON [PRIMARY]
11 GO

```

Program 4.10 Perintah SQL untuk membuat tabel komponen kependudukan beserta constraintnya

Program 4.10 merupakan program sql untuk membuat tabel komponen kependudukan dengan *field-fieldnya* ID_Komponen_Kependudukan, dan Nama Komponen Kependudukan. Serta *constraint* PK Komponen Kependudukan

```

1  CREATE TABLE [dbo].[Kependudukan] (
2  [ID_Kependudukan] [int] IDENTITY (1, 1) NOT FOR REPLICATION NOT
   NULL ,
3  [ID_Komponen_Kependudukan] [int] NOT NULL ,
4  [Tahun] [int] NULL ,
5  [Nilai] [float] NULL
6  ) ON [PRIMARY]
7  GO
8  ALTER TABLE [dbo].[Kependudukan] WITH NOCHECK ADD
9  CONSTRAINT [PK_Item_Komponen_Kependudukan] PRIMARY KEY CLUSTERED
10 (
11 [ID_Kependudukan]
12 ) ON [PRIMARY]
13 GO
14 ALTER TABLE [dbo].[Kependudukan] ADD
15 CONSTRAINT [FK_Kependudukan_Komponen_Kependudukan] FOREIGN KEY
16 (
17 [ID_Komponen_Kependudukan]
18 ) REFERENCES [dbo].[Komponen_Kependudukan] (
19 [ID_Komponen_Kependudukan]
20 ) ON DELETE CASCADE ON UPDATE CASCADE
21 GO

```


2. PEMBUATAN PROSES APLIKASI

Terdapat beberapa fungsi yang digunakan dalam pembuatan aplikasi untuk perencanaan trayek Angkutan Kota, yang dibagi ke dalam 2 bagian program utama, yaitu untuk masukan basis data dan perencanaan trayek angkutan kota, yang selanjutnya akan dijelaskan pada sub-bab berikut.

2. 1. Masukan Basis-Data

Masukan basis-data merupakan proses untuk memasukkan data-data yang dibutuhkan dalam proses aplikasi perencanaan trayek angkutan kota. Terdapat beberapa jenis masukan data seperti yang telah dijelaskan pada sub-bab perencanaan proses, yang digolongkan ke dalam empat jenis proses masukan data, yaitu masukan data kependudukan, masukan data trayek, masukan data survey, dan masukan data wilayah.

2. 1. 1. Proses Masukan Data Kependudukan

Proses masukan data kependudukan terdiri dari proses masukan data jumlah penduduk, proses masukan data jumlah pendapatan, dan proses masukan data jumlah kendaraan.

2. 1. 1. 1. Proses Masukan Data Jumlah Penduduk

Proses masukan data jumlah penduduk adalah proses memasukkan data jumlah penduduk tiap tahun ke dalam tabel kependudukan. Data yang dimasukkan tersebut adalah tahun dan jumlah penduduk dimana dalam pembuatan proses

DataSource 'DataModuleSPK.Data_Kependudukan' dengan nama *DataField*-nya 'Nilai'.

4. 2. 1. 2. Proses Masukan Data Trayek

Proses masukan data trayek adalah proses memasukkan data trayek ke dalam tabel trayek, dengan data nama trayek yang dimasukkan ke dalam *DataSource* DataModuleSPK.Data_Trayek dengan nama *DataField*-nya Nama_Trayek (gambar 4.3) dan tarif kendaraan ke dalam *DataSource* DataModuleSPK.Data_Trayek dengan nama *DataField*-nya Tarif (gambar 4.4).

DataField **Nama_Trayek**
 DataSource **DataModuleSPK.Data_Trayek**

Gambar 4.3 Masukan Data Ke *DataField* Nama_Trayek dan *DataSource* DataModuleSPK.Data_Trayek

DataField **Tarif**
 DataSource **DataModuleSPK.Data_Trayek**

Gambar 4.4 Masukan Data Ke *DataField* Trayek dan *DataSource* DataModuleSPK.Data_Trayek

Selain itu pada masukan data trayek juga terdapat masukan rute trayek ke dalam tabel rute, dengan data no_rute yang dimasukkan ke dalam *DataSource* DataModuleSPK.Data_Rute dengan nama *DataField*-nya no_urut (gambar 4.5) dan link yang dilalui ke dalam *DataSource* DataModuleSPK.Data_Rute dengan nama *DataField*-nya id_link (gambar 4.6).

DataField **no_urut**
 DataSource **DataModuleSPK.Data_Rute**

Gambar 4.5 Masukan Data Ke *DataField* no_urut dan *DataSource*

4. 2. 1. 3. Proses Masukan Data Survey HI

Proses masukan data survey HI merupakan proses memasukkan data survey ke dalam basis-data SPK. Terdapat beberapa jenis masukan data survey HI, yaitu proses masukan survey HI, proses masukan asal tujuan perjalanan, proses masukan sampel HI.

4. 2. 1. 3. 1. Proses Masukan Survey HI

Proses masukan survey HI merupakan proses memasukkan data survey HI untuk menyimpan data keluarga tiap zona ke dalam tabel *total_data_keluarga*. Data-data tersebut adalah zona yang dimasukkan ke dalam *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga* dengan nama *DataField*-nya *ID_Zona* (gambar 4.7), total jumlah anggota keluarga ke dalam *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga* dengan nama *DataField*-nya *Total_Anggota_Keluarga* (gambar 4.8), jumlah total pendapatan ke dalam *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga* dengan nama *DataField*-nya *Total_Pendapatan* (gambar 4.9), jumlah total kendaraan ke dalam *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga* dengan nama *DataField*-nya *Total_Kendaraan* (gambar 4.10).

DataField ID_Zona

DataSource DataModuleSPK.Data_Total_Data_Keluarga

Gambar 4.7 Masukan Data Ke *DataField* *ID_Zona* dan *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga*

DataField Total_Anggota_Keluarga

DataSource DataModuleSPK.Data_Total_Data_Keluarga

Gambar 4.8 Masukan Data Ke *DataField* *Total_Anggota_Keluarga* dan *DataSource* *DataModuleSPK.Data_Total_Data_Keluarga*

DataModuleSPK.Data_Total_Data_Keluarga

DataField **Total_Kendaraan**
 DataSource **DataModuleSPK.Data_Total_Data_Keluarga**

Gambar 4.10 Masukan Data Ke *DataField* **Total_Kendaraan** dan *DataSource* **DataModuleSPK.Data_Total_Data_Keluarga**

4. 2. 1. 3. 2. Proses Masukan Asal Tujuan Perjalanan

Proses masukan asal tujuan perjalanan adalah proses memasukkan data survey perjalanan antar zona ke dalam tabel ‘Asal Tujuan Perjalanan’. Data-data tersebut adalah zona asal yang dimasukkan ke dalam *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan** dengan nama *DataField*-nya **Zona_Asal** (gambar 4.11), zona tujuan ke dalam *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan** dengan nama *DataField*-nya **Zona_Tujuan** (gambar 4.12), jumlah perjalanan ke dalam *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan** dengan nama *DataField*-nya **Jumlah_Perjalanan** (gambar 4.13), jumlah perjalanan menggunakan angkutan kota ke dalam *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan** dengan nama *DataField*-nya **Jumlah_Perjalanan_Angkutan** (gambar 4.14).

DataField **Zona_Asal**
 DataSource **DataModuleSPK.Data_Asal_Tujuan_Perjalanan**

Gambar 4.11 Masukan Data Ke *DataField* **Zona_Asal** dan *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan**

DataField **Zona_Tujuan**
 DataSource **DataModuleSPK.Data_Asal_Tujuan_Perjalanan**

Gambar 4.12 Masukan Data Ke *DataField* **Zona_Tujuan** dan *DataSource* **DataModuleSPK.Data_Asal_Tujuan_Perjalanan**

DataField **Jumlah_Perjalanan**
 DataSource **DataModuleSPK.Data_Asal_Tujuan_Perjalanan**

4. 2. 1. 3. 3. Proses Masukan Sampel Survey

Proses masukan sampel survey adalah proses memasukkan data jumlah sampel yang diambil dalam melakukan survey HI, dan data tersebut adalah zona yang dimasukkan ke dalam *DataSource* *DataModuleSPK.Data_Sampel_HI* dengan nama *DataField*-nya *ID_Zona* (gambar 4.15), jumlah keluarga yang disurvey ke dalam *DataSource* *DataModuleSPK.Data_Sampel_HI* dengan nama *DataField*-nya *Jumlah_Keluarga_Disurvey* (gambar 4.16), jumlah seluruh keluarga ke dalam *DataSource* *DataModuleSPK.Data_Sampel_HI* dengan nama *DataField*-nya *Jumlah_Seluruh_Keluarga* (gambar 4.17), jumlah keluarga yang menolak diwawancara ke dalam *DataSource* *DataModuleSPK.Data_Sampel_HI* dengan nama *DataField*-nya *Keluarga_Menolak_Diwawancara* (gambar 4.18), jumlah rumah kosong ke dalam *DataSource* *DataModuleSPK.Data_Sampel_HI* dengan nama *DataField*-nya *Jumlah_Rumah_Kosong* (gambar 4.19).

DataField **ID_Zona**
☒ *DataSource* **DataModuleSPK.Data_Sampel_HI**

Gambar 4.15 Masukan Data Ke *DataField* ID_Zona dan *DataSource* DataModuleSPK.Data_Sampel_HI

DataField **Jumlah_Keluarga_Disurvey**
☒ *DataSource* **DataModuleSPK.Data_Sampel_HI**

Gambar 4.16 Masukan Data Ke *DataField* Jumlah_Keluarga_Disurvey dan *DataSource* DataModuleSPK.Data_Sampel_HI

DataField **jumlah_seluruh_keluarga**
☒ *DataSource* **DataModuleSPK.Data_Sampel_HI**

Gambar 4.17 Masukan Data Ke *DataField* Jumlah_Seluruh_Keluarga dan *DataSource* DataModuleSPK.Data_Sampel_HI

DataField **jumlah_rumah_kosong**
 DataSource **DataModuleSPK.Data_Sampel_HI**

Gambar 4.19 Masukan Data Ke *DataField* Jumlah_Rumah_Kosong dan *DataSource* DataModuleSPK.Data_Sampel_HI

4. 2. 1. 4. Proses Masukan Data Wilayah

Proses masukan data wilayah adalah proses memasukkan data-data yang berhubungan dengan kewilayahan Kota Mojokerto seperti masukan data zona, jalan, kelurahan, wilayah zona,

4. 2. 1. 4. 1. Masukan Data Zona

Proses masukan data zona adalah proses memasukkan data zona-zona yang ada di kota Mojokerto ke dalam tabel zona, data tersebut adalah nama zona yang dimasukkan ke dalam *DataSource* DataModuleSPK.Data_Zona dengan nama *DataField*-nya nama_zona, seperti dapat dilihat pada gambar 4.20.

DataField **nama_zona**
 DataSource **DataModuleSPK.Data_Zona**

Gambar 4.20 Masukan Data Ke *DataField* Nama_Zona dan *DataSource* DataModuleSPK.Data_Zona

4. 2. 1. 4. 2. Masukan Data Jalan

Proses masukan data jalan adalah proses memasukkan data jalan-jalan yang ada di kota Mojokerto ke dalam tabel Jalan, data tersebut adalah nama jalan yang dimasukkan ke dalam *DataSource* DataModuleSPK.Data_Jalan dengan nama *DataField*-nya nama_jalan, seperti dapat dilihat pada gambar 4.21.

DataField **nama_jalan**
 DataSource **DataModuleSPK.Data_Jalan**

kelurahan yang dimasukkan ke dalam *DataSource* *DataModuleSPK.Data_Kelurahan* dengan nama *DataField*-nya *nama_kelurahan*, seperti dapat dilihat pada gambar 4.22.



Gambar 4.22 Masukan Data Ke *DataField* Nama_Kelurahan dan *DataSource* *DataModuleSPK.Data_Kelurahan*

4. 2. 1. 4. 4. Masukan Data Wilayah Zona

Masukan data wilayah zona adalah proses memasukkan pembagian wilayah-wilayah yang ada di kota Mojokerto ke dalam zona-zona, dimana wilayah wilayah tersebut didasarkan pada kelurahan yang ada di kota Mojokerto. Zona yang ada di kota Mojokerto diambil dari tabel zona dan ditampilkan dalam bentuk *combo box* dengan menggunakan daftar program seperti pada gambar 4.14 baris 4 sampai 10. Kemudian setiap kali terdapat *record* baru yang masuk ke tabel wilayah zona maka secara otomatis *ID_Zona* yang dipilih dari *combo box* juga akan ikut masuk ke tabel wilayah_zona seperti dapat dilihat pada gambar 4.14 baris 13 sampai 16.

```

1 procedure TFormWilayahZona.FormActivate(Sender: TObject);
2 begin
3     DataModuleSPK.Tabel_Wilayah_Zona.OnNewRecord:=
4     Tabel_Wilayah_ZonaNewRecord;
5     ComboBox1.Clear;
6     DataModuleSPK.Tabel_Zona.First;
7     while not DataModuleSPK.Tabel_Zona.Eof do
8     begin
9         ComboBox1.Items.Add(DataModuleSPK.Tabel_Zona.FieldValues['nama_zona'
10         ]);
11         DataModuleSPK.Tabel_Zona.Next;
12     end;
13     DataModuleSPK.Tabel_Wilayah_Zona.Close;
14 end;
15 procedure TFormWilayahZona.Tabel_Wilayah_ZonaNewRecord(DataSet:
16 TDataSet);

```


DataModuleSPK.Data_Kelurahan dalam bentuk *combo box* (gambar 4.23). Kemudian kelurahan yang dipilih akan dimasukkan ke dalam *DataSource* DataModuleSPK.Data_Wilayah_Zona dengan nama *DataField*-nya *id_kelurahan* (gambar 4.24).

```

KeyField  id_kelurahan
Left      168
ListField nama_kelurahan
ListFieldInc 0
ListSource DataModuleSPK.Data_Kelurahan

```

Gambar 4.23 Mengambil Data Nama_kelurahan Dengan KeyField ID_Kelurahan Dari *ListSource* DataModuleSPK.Data_Kelurahan

```

DataField id_kelurahan
DataSource DataModuleSPK.Data_Wilayah_Zona

```

Gambar 4.24 Masukan Data Ke *DataField* ID_Kelurahan dan *DataSource* DataModuleSPK.Data_Wilayah_Zona

4. 2. 1. 4. 5. Masukan Data Link

Masukan data link adalah proses memasukkan link-link yang ada di kota Mojokerto ke dalam tabel link. Data-data tersebut antara lain adalah jalan, dimana data jalan diambil dari tabel jalan dan ditampilkan dalam bentuk *combo box* dengan menggunakan daftar program seperti pada gambar 4.15 baris 4 sampai 10. Kemudian setiap kali terdapat *record* baru yang masuk ke tabel link maka secara otomatis ID_Jalan yang dipilih dari *combo box* juga akan ikut masuk ke tabel link seperti dapat dilihat pada gambar 4.15 baris 13 sampai 16.

Selain data jalan juga ditampilkan *ListField* nama_node dengan KeyField-nya *id_node* yang diambil dari *ListSource* DataModuleSPK.Data_Node dalam bentuk *combo box* (gambar 4.26). Kemudian node yang dipilih akan dimasukkan

DataModuleSPK.Data_Link dengan nama *DataField*-nya arus_lalu_lintas (gambar 4.30), kapasitas ke dalam *DataSource* DataModuleSPK.Data_Link dengan nama *DataField*-nya kapasitas (gambar 4.31), kecepatan ke dalam *DataSource* DataModuleSPK.Data_Link dengan nama *DataField*-nya kecepatan (gambar 4.32).

```

1  procedure TFormLink.FormActivate(Sender: TObject);
2  begin
3      DataModuleSPK.Tabel_Link.OnNewRecord := Tabel_LinkNewRecord;
4      ComboBox1.Clear;
5      DataModuleSPK.Tabel_Jalan.First;
6      while not DataModuleSPK.Tabel_Jalan.Eof do
7          begin
8              ComboBox1.Items.Add(DataModuleSPK.Tabel_Jalan.FieldValues
9                  ['Nama_Jalan']);
10             DataModuleSPK.Tabel_Jalan.Next;
11         end;
12     end;
13     DataModuleSPK.Tabel_Link.Close;
14 end;
15 procedure TFormLink.Tabel_LinkNewRecord(DataSet: TDataSet);
16 begin
17     DataModuleSPK.Tabel_Link.FieldValues['ID_Jalan'] :=
18         DataModuleSPK.Tabel_Jalan.Lookup('Nama_Jalan',
19             ComboBox1.Text, 'ID_Jalan');
20 end;

```

Program 4.15 Masukan Data Jalan Ke Tabel Link

DataField **Nama_Link**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.25 Masukan Data Ke DataField Nama_Link dan DataSource DataModuleSPK.Data_Link

KeyField **id_node**
 Left **152**
 ListField **nama_node**
 ListFieldInc 0
 ListSource **DataModuleSPK.Data_Node**

Gambar 4.26 Mengambil Data Nama Node Dengan KeyField ID Node Dari ListSource

DataField **id_node_akhir**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.28 Masukan Data Ke *DataField* ID_Node_Akhir dan *DataSource*
DataModuleSPK.Data_Link

DataField **panjang**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.29 Masukan Data Ke *DataField* Panjang dan *DataSource*
DataModuleSPK.Data_Link

DataField **arus_lalu_lintas**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.30 Masukan Data Ke *DataField* Arus_Lalu_Lintas dan *DataSource*
DataModuleSPK.Data_Link

DataField **kapasitas**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.31 Masukan Data Ke *DataField* Kapasitas dan *DataSource*
DataModuleSPK.Data_Link

DataField **kecepatan**
 DataSource **DataModuleSPK.Data_Link**

Gambar 4.32 Masukan Data Ke *DataField* Kecepatan dan *DataSource*
DataModuleSPK.Data_Link

4. 2. 1. 4. 6. Masukan Data Node

Masukan data node adalah proses memasukkan data node yang ada di kota Mojokerto ke dalam tabel node. Data-data tersebut adalah zona yang diambil dari tabel zona dan ditampilkan dalam bentuk *combo box* seperti dapat dilihat pada daftar program 4.16 baris 4 sampai 8, dan selanjutnya semua data yang ditampilkan disesuaikan dengan zona yang dipilih seperti dapat dilihat pada


```

1  procedure TFormDataNode.FormActivate(Sender: TObject);
2  begin
3      DataModuleSPK.Tabel_Zona.First;
4      while not DataModuleSPK.Tabel_Zona.Eof do
5          begin
6              ComboBox1.Items.Add(DataModuleSPK.Tabel_Zona.FieldValue
7                  s['nama_zona']);
8              DataModuleSPK.Tabel_Zona.Next;
9          end;
10 end;
11 procedure TFormDataNode.ComboBox1Change(Sender: TObject);
12 begin
13     if ComboBox1.Text = '' then
14         begin
15             DataModuleSPK.Tabel_Node.Close;
16             Edit1.Enabled := False;
17             ComboBox2.Enabled := False;
18             Button1.Enabled := False;
19             Button2.Enabled := False;
20         end else
21         begin
22             DataModuleSPK.Tabel_Node.Open;
23             DataModuleSPK.Tabel_Node.Filtered := False;
24             DataModuleSPK.Tabel_Node.Filter := 'ID_Zona = ' +
25                 IntToStr(DataModuleSPK.Tabel_Zona.Lookup('Nama_Zona',
26                     ComboBox1.Text, 'ID_Zona'));
27             DataModuleSPK.Tabel_Node.Filtered := True;
28             Edit1.Enabled := True;
29             ComboBox2.Text := '';
30             ComboBox2.Enabled := True;
31             Button1.Enabled := True;
32             Button2.Enabled := True;
33         end;
34     end;
35 end;
36 procedure TFormDataNode.ComboBox2Change(Sender: TObject);
37 begin
38     if ComboBox2.Text = 'Node Pusat' then Node := 1
39     else if ComboBox2.Text = 'Bukan Node Pusat' then Node :=
40         0;
41 end;
42 procedure TFormDataNode.Button1Click(Sender: TObject);
43 begin
44     DataModuleSPK.Tabel_Node.Append;
45     DataModuleSPK.Tabel_Node.FieldValues['id_zona']:=
46         DataModuleSPK.Tabel_Zona.Lookup('nama_zona',
47             ComboBox1.Text, 'id_zona');
48     DataModuleSPK.Tabel_Node.FieldValues['nama_node'] :=
49         Edit1.Text;
50     DataModuleSPK.Tabel_Node.FieldValues['node_pusat'] :=
51         Node;
52     DataModuleSPK.Tabel_Node.Post;

```

4. 2. 2. Proses Perencanaan Trayek

Proses perencanaan trayek merupakan proses untuk merencanakan trayek angkutan kota yang sudah ada, berdasarkan pada data-data yang telah tersedia pada basis-data 'PERENCANAAN TRAYEK'. Terdapat beberapa proses di dalamnya yang akan dijelaskan pada sub-bab berikut.

4. 2. 2. 1. Peramalan Matrik Asal Tujuan Perjalanan

Peramalan matrik asal tujuan perjalanan adalah salah satu bagian program Perencanaan Trayek Angkutan Kota yang berfungsi untuk melakukan peramalan perjalanan dari zona satu ke zona lainnya. Di dalamnya terdiri dari banyak unit yang mempunyai fungsi dan kegunaan masing-masing. Pada sub-bab berikut akan dijelaskan masing-masing unit tersebut beserta daftar programnya.

4. 2. 2. 1. 1. Pencarian Persamaan Regresi

Persamaan regresi digunakan untuk peramalan data perjalanan pada suatu tahun rencana. Persamaan regresi didapatkan dari fungsi regresi dengan parameter keluaran koefisien-koefisien yang akan dijadikan koefisien persamaan regresi, dan parameter masukan data-data yang akan diregresikan yang berasal dari tabel tempat data tersebut disimpan beserta dengan nama-nama *field*-nya. Program untuk mendapatkan persamaan regresi terdiri dari beberapa fungsi dan prosedur yang akan dijelaskan pada sub-bab berikut.

4. 2. 2. 1. 1. 1. Prosedur Multiple_Matrix

Prosedur Multiple_Matrix digunakan untuk mencari nilai konstanta dan koefisien persamaan regresi, yang didapat dari perkalian antara invers matriks A dengan matriks g (baris 6 sampai 12), seperti dinyatakan dalam persamaan 2.13

2.2.1.1.4. Fungsi Taksiran

Fungsi Taksiran digunakan untuk menghitung taksiran tak bias untuk variansi σ^2 . Baris 3 sampai baris 12 merupakan perintah sql yang digunakan untuk mengambil nilai kuadrat dari pengurangan antara y data dan y hasil perhitungan. Kemudian nilai tersebut dimasukkan ke dalam suatu rumus menghitung nilai taksiran seperti dinyatakan dalam persamaan 2.16 (baris 14). Daftar program fungsi untuk menghitung nilai taksiran tersebut dapat dilihat pada gambar 4.20.

```

1  function Taksiran(n, k : Integer; bx : TVectorExt): Extended;
2  begin
3      Database.SQL.Add('SELECT SUM((' + Tabel + '.' + Y + ' - (' +
        FloatToStr(bx[0]) + ' + ');
4      Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' + ');
5      Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' + ');
6      Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' + ');
7      Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + ')) *
        ');
8      Database.SQL.Add('(' + Tabel + '.' + Y + ' - (' + FloatToStr(bx[0]) + '
        + ');
9      Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' + ');
10     Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' + ');
11     Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' + ');
12     Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + '))) AS
        Result FROM ' + Tabel);
13     Database.Open;
14     Result := Database.FieldValues['Result'] / (n - k - 1);
15     Database.Close;
16     Database.SQL.Clear;
17 end;
```

Program 4.20 Fungsi Taksiran

2.2.1.1.5. Fungsi Uji-T

Fungsi ini digunakan untuk menghitung nilai t (baris 9 atau 11) yang digunakan dalam uji t-test untuk menguji signifikan nilai koefisien regresi dan

ibandingkan dengan nilai t hasil perhitungan (sub bab 4.2.1.3.3. dan sub bab 4.2.1.3.5.). Daftar program untuk fungsi melihat tabel t dapat dilihat pada program 4.22.

```

1 function Uji_t(k : Integer; s : Extended; c : TMatrix; bx : TVectorExt):
  TVectorExt;
2 var
3   i : Integer;
4   t : TVectorExt;
5 begin
6   SetLength(t,k+1);
7   for i := 0 to k do
8     if i = 0 then
9       t[i] := bx[i] / sqrt(s * c[i,i])
10    else
11      t[i] := bx[x[i-1]] / sqrt(s * c[i,i]);
12  Result := t;
13 end;
```

Program 4.21 Fungsi Uji T

```

1 function Tabel_t(p : Extended; df : Integer): Extended;
2 begin
3   Database.SQL.Add('SELECT Tabel_t.t FROM Tabel_t WHERE (((Tabel_t.p) = '
4     + FloatToStr(p) + ') AND ((Tabel_t.df) = ' + IntToStr(df) + '))');
5   Database.Open;
6   Result := Database.FieldValues['t'];
7   Database.Close;
8   Database.SQL.Clear;
```

Program 4.22 Fungsi Tabel T

4.2.2.1.1.7. Fungsi Koefisien Determinasi

Fungsi ini digunakan untuk menghitung nilai koefisien determinasi dari suatu persamaan regresi. Dapat dilihat pada program 4.23 baris 3 sampai 13 merupakan perintah sql yang digunakan untuk mengambil nilai kuadrat dari

Untuk kasus persamaan regresi linier berganda dilakukan koreksi terhadap nilai koefisien determinasi yang diperoleh (baris 3), seperti dinyatakan pada persamaan 2.14. Daftar program dari fungsi ini dapat dilihat pada program 4.24.

```

1  function Koef_Det(bx : TVectorExt): Extended;
2  begin
3      Database.SQL.Add('SELECT SUM((' + FloatToStr(bx[0]) + ' + ');
4      Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' + ');
5      Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' + ');
6      Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' + ');
7      Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + ' -
      ');
8      Database.SQL.Add(FloatToStr(AvgY) + ') * ((' + FloatToStr(bx[0]) + ' +
      ');
9      Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' + ');
10     Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' + ');
11     Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' + ');
12     Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + ' -
      ');
13     Database.SQL.Add(FloatToStr(AvgY) + ')) AS Result1, ');
14     Database.SQL.Add('SUM((' + Tabel + '.' + Y + ' - ' + FloatToStr(AvgY) +
      ') * (');
15     Database.SQL.Add(Tabel + '.' + Y + ' - ' + FloatToStr(AvgY) + ')) AS
      Result2 FROM ' + Tabel);
16     Database.Open;
17     Result := Database.FieldValues['Result1'] /
      Database.FieldValues['Result2'];
18     Database.Close;
19     Database.SQL.Clear;
20 end;

```

Program 4.23 Fungsi Koefisien Determinasi

```

1  function Koef_Det_koreksi(R : Extended; n, k : Integer): Extended;
2  begin
3      Result := (R - k / (n - 1)) * ((n - 1) / (n - k - 1));
4  end;

```

Program 4.24 Fungsi Koefisien Determinasi Koreksi

yang diperoleh tersebut dimasukkan ke dalam sebuah persamaan untuk mendapatkan nilai F (baris 15).

```

1 function Uji_F(k : Integer; s : Extended; bx : TVectorExt): Extended;
2 begin
3     Database.SQL.Add('SELECT SUM((' + FloatToStr(bx[0]) + ' + ' + ');
4     Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' +
5     Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' +
6     Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' +
7     Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + ' + ' -
8     Database.SQL.Add(FloatToStr(AvgY) + ') * ((' + FloatToStr(bx[0]) + ' +
9     Database.SQL.Add(FloatToStr(bx[1]) + ' * ' + Tabel + '.' + X1 + ' +
10    Database.SQL.Add(FloatToStr(bx[2]) + ' * ' + Tabel + '.' + X2 + ' +
11    Database.SQL.Add(FloatToStr(bx[3]) + ' * ' + Tabel + '.' + X3 + ' +
12    Database.SQL.Add(FloatToStr(bx[4]) + ' * ' + Tabel + '.' + X4 + ' + ' -
13    Database.SQL.Add(FloatToStr(AvgY) + ')) AS Result FROM ' + Tabel);
14    Database.Open;
15    Result := (Database.FieldValues['Result'] / k) / s;
16    Database.Close;
17    Database.SQL.Clear;
18 end;

```

Program 4.25 Fungsi Uji F

2.2.1.1.9. Fungsi Tabel F

Fungsi ini digunakan untuk melihat nilai F pada tabel-F dengan derajat keyakinan (k , $n-k-1$) dan $\alpha = 5\%$ (baris 3 sampai baris 5) yang nantinya akan dibandingkan dengan nilai F hasil perhitungan (sub bab 4.2.1.3.8.). Daftar program untuk fungsi melihat tabel F dapat dilihat pada program 4.26.

```

1 function Tabel_F(p : Extended; df1, df2 : Integer): Extended;

```


4. 2. 2. 1. 1. 10. **Fungsi Inverse**

Fungsi ini digunakan untuk mendapatkan nilai invers dari suatu matriks. Langkah awal yang dilakukan adalah mencari nilai untuk kolom dan baris yang sama, dan dimasukkan ke dalam variabel *ps* (baris 9), kemudian mengganti nilai sel untuk kolom dan baris yang sama tersebut dengan nilai 1 (baris 10). Untuk semua sel pada barisnya, nilai masing-masing sel tersebut dibagi dengan *ps* (baris 15). Kemudian dilihat baris lainnya. Sel pada baris lain dengan kolom sama dengan kolom asal *ps*, nilainya dimasukkan ke dalam suatu variabel *pn* (baris 21), dan nilai sel tersebut diganti dengan nilai 0 (baris 22). Kemudian untuk semua sel pada baris tersebut dikurangi dengan nilai perkalian antara *pn* dengan nilai pada baris asal *ps* dengan kolom yang sama dengan sel yang akan diganti nilainya tersebut (baris 25). Kemudian proses tersebut diulang untuk baris yang tersisa. dari fungsi ini dapat dilihat pada program 4.27.

4. 2. 2. 1. 1. 11. **Fungsi regresi**

Fungsi *regresi* adalah fungsi utama dari proses untuk mencari suatu persamaan regresi, dimana fungsi ini akan memanggil fungsi dan prosedur yang telah dijelaskan sebelumnya pada sub-bab 4.2.1.3.1 sampai 4.2.1.3.10 sesuai dengan kebutuhan. Langkah awal yang dilakukan untuk mendapatkan persamaan regresi dalam fungsi *regresi* ini adalah dengan melihat hubungan korelasi antara sesama variabel bebas, dan antara variabel bebas dengan variabel tidak bebasnya (baris 6 program 4.28). Antara variabel bebas dengan variabel tidak bebas tidak boleh sama dengan nol, atau harus ada korelasi. Bila tidak ada korelasi atau korelasinya sama dengan nol maka variabel bebas tersebut tidak dapat diterima (baris 12 program 4.28). Selain itu variabel bebas yang ada juga harus

bebasnya paling tinggi. Maka langkah sebelumnya yang dilakukan adalah mengurutkan korelasi antara variabel bebas dengan variabel tidak bebasnya dari yang paling besar sampai dengan yang paling kecil (baris 2 sampai 9 program 4.29) . Kemudian setelah itu melihat korelasi antar variabel tidak bebasnya, dengan asumsi awal bahwa tidak ada korelasi adalah untuk nilai korelasi di bawah 0.2 (baris 14 sampai 18 program 4.29).

```

1  procedure Inverse_Matrix(a : TMatrix; var x: TMatrix);
2  var
3      ps,pn : real;
4      i,j,k,y : Integer;
5  begin
6      x := a;
7      for i:=0 to High(a) do
8          begin
9              ps:=x[i,i];
10             x[i,i]:=1;
11             if ps<>0 then
12                 begin
13                     for j:=0 to High(a) do
14                         begin
15                             x[i,j]:=x[i,j]/ps;
16                         end;
17                     for k:=0 to High(a) do
18                         begin
19                             if k<>i then
20                                 begin
21                                     pn:=x[k,i];
22                                     x[k,i]:=0;
23                                     for j:=0 to High(a) do
24                                         begin
25                                             x[k,j]:=x[k,j]-pn*x[i,j];
26                                         end;
27                                     end;
28                                 end;
29                             end else
30                                 begin
31                                     for y:=0 to High(a) do
32                                         for j:=0 to High(a) do
33                                             x[y,j]:=0;
34                                         end;
35                                     end;
36                                 end;

```

```

1  SetLength(r,5,5);
2  SetLength(t,5,5);
3  for i := 0 to 4 do
4      for j := 0 to 4 do
5          begin
6              r[i,j] := Korelasi(n, svv[i,j], sv[i], sv[j], svv[i,i], svv[j,j]);
7              t[i,j] := Uji_t0(n, r[i,j]);
8              if (i > 0) and (j = 0) then
9                  begin
10                     p[i].x := i;
11                     p[i].r := r[i,j];
12                     p[i].Accept := (abs(r[i,j]) > 0) and ((r[i,j] = 1) or
13                                     (abs(t[i,j]) > Tabel_t(0.025, n-2)));
14                 end;
15             end;
16         end;
17     end;

```

Program 4.28 Unit Regresi (1)

Setelah diketahui jumlah variabel bebas yang diterima, maka dilakukan perulangan sebanyak variabel bebas yang ada untuk mencari persamaan regresi dengan penghilangan satu buah variabel bebas setiap kali perulangan. Variabel bebas yang dihilangkan adalah variabel yang mempunyai korelasi paling kecil dengan variabel tidak bebasnya.

Dalam mencari persamaan regresi tersebut, pertama kali dicari nilai matriks A yang diperoleh dari basis-data yang ada (baris 5 sampai baris 17 program 4.31). Dicari pula nilai G (baris 7 dan baris 18 sampai 21 program 4.31) yang juga diambil dari suatu basis-data. Selanjutnya dapat dicari nilai B yang merupakan koefisien dan konstanta dari persamaan regresi yang dihasilkan (baris 24 program 4.31), dengan mengalikan antara invers matriks A dan matriks g dengan memanggil fungsi `multiple_matrix`. Kemudian setelah mendapatkan koefisien variabel bebas, nilai dari koefisien variabel bebas tersebut dimasukkan ke dalam sebuah variabel *array* Bx. Dapat dilihat pada program 4.32.

pada tabel untuk harga mutlaknya dan F yang dihasilkan harus lebih besar dari nilai F pada tabel (baris 6 sampai 10 program 4.33). Sehingga didapatkan persamaan regresi dengan banyaknya variabel bebas yang bervariasi.

```

1  {*** Mengurut variabel bebas X berdasarkan korelasi dari besar ke kecil
    ***}
2  for i := 1 to 3 do
3    for j := i + 1 to 4 do
4      if abs(p[i].r) < abs(p[j].r) then
5        begin
6          p[0] := p[i];
7          p[i] := p[j];
8          p[j] := p[0];
9        end;
10 p[0].x := 0;
11 p[0].r := 0;
12 p[0].Accept := False;
13 {*** Memilih variabel bebas X berdasarkan syarat hasil korelasi <= 0.2
    *** dibawah 0.2 dianggap tidak ada korelasi, batasan asumsi}
14 for i := 1 to 3 do
15   if p[i].Accept then
16     for j := i + 1 to 4 do
17       if p[j].Accept then
18         p[j].Accept := (abs(r[p[i].x,p[j].x]) <= 0.2);

```

Program 4.29 Unit Regresi (2)

```

1  {*** Mendapatkan urutan variabel bebas X yang diterima ***}
2  SetLength(x,4);
3  k := 0;
4  for i := 1 to 4 do
5    if p[i].Accept then
6      begin
7        x[i-1] := p[i].x;
8        Inc(k);
9      end;

```

Program 4.30 Unit Regresi (3)

```

1      SetLength(A,k+1,k+1);
2      SetLength(invA,k+1,k+1);
3      SetLength(g,k+1,1);
4      SetLength(B,k+1,1);
5      for i := 0 to k do
6      begin
7          for j := 0 to k do
8              if i = 0 then
9                  if j = 0 then
10                     A[i,j] := n
11                 else
12                     A[i,j] := sv[x[j-1]]
13             else
14                 if j = 0 then
15                     A[i,j] := sv[x[i-1]]
16                 else
17                     A[i,j] := svv[x[i-1],x[j-1]];
18             if i = 0 then
19                 g[i,0] := sv[0]
20             else
21                 g[i,0] := svv[0,x[i-1]];
22         end;
23     Inverse_Matrix(A, invA);
24     Multiple_Matrix(invA, g, B);

```

Program 4.31 Unit regresi (4)

```

1  SetLength(d[k].Bx,5);
2  j := 1;
3  d[k].Bx[0] := B[0,0];
4  for i := 1 to k do
5      if x[i-1] <> 0 then
6      begin
7          d[k].Bx[x[i-1]] := B[j,0];
8          Inc(j);
9      end;

```

Program 4.32 Unit Regresi (5)

```

1  d[k].R := Koef_Det(d[k].Bx);
2  d[k].R_koreksi := Koef_Det_koreksi(d[k].R, n, k);
3  SetLength(t_hit,k+1);
4  t_hit := Uj_t(k, Taksiran(n, k, d[k].Bx), InvA, d[k].Bx);
5  F_hit := Uji_F(k, Taksiran(n, k, d[k].Bx), d[k].Bx);

```

```

1  SetLength(d[0].Bx, 5);
2  for i := 1 to 4 do
3      if d[i].Accept then
4          if d[i].R_koreksi > d[0].R_koreksi then d[0] := d[i];
5  B0 := d[0].Bx[0];
6  B1 := d[0].Bx[1];
7  B2 := d[0].Bx[2];
8  B3 := d[0].Bx[3];
9  B4 := d[0].Bx[4];

```

Program 4.34 Unit Regresi (7)

2. 2. 1. 2. Peramalan Data Kependudukan dan Bangkitan Perjalanan

Dalam peramalan data bangkitan perjalanan diperlukan data-data variabel bebas yang mendukung persamaan regresi untuk meramalkan bangkitan perjalanan pada suatu tahun rencana. Variabel bebas yang akan digunakan dalam persamaan regresi disesuaikan dengan persamaan regresi yang dihasilkan. Koefisien b_1 pada persamaan regresi linier merupakan koefisien untuk data variabel bebas jumlah penduduk, b_2 untuk jumlah pendapatan, dan b_3 untuk jumlah kendaraan. Untuk mendapatkan jumlah perjalanan pada suatu tahun rencana maka variabel-variabel bebasnya harus diramalkan untuk suatu tahun rencana juga, dimana untuk meramalkan tersebut dengan menggunakan rumus tingkat pertumbuhan. Dalam rumus tingkat pertumbuhan tersebut terdapat faktor pertumbuhan.

Faktor pertumbuhan digunakan untuk menghitung nilai data kependudukan, seperti jumlah penduduk, jumlah pendapatan, dan jumlah kendaraan pada suatu tahun rencana. Faktor pertumbuhan disini adalah faktor pertumbuhan rata-rata dari data kependudukan untuk periode beberapa tahun. Daftar program untuk menghitung faktor pertumbuhan tersebut dapat dilihat pada program 4.35. Faktor pertumbuhan didapat dari data tahun Y dikurangi data tahun

Setelah faktor pertumbuhan untuk masing-masing data kependudukan diperoleh, selanjutnya data kependudukan untuk tahun rencana dapat dihitung (program 4.36 baris 3 sampai baris 5), sehingga diperoleh jumlah perjalanan pada suatu tahun rencana (program 4.36 baris 6). Selengkapnya untuk peramalan data perjalanan dapat dilihat pada program 4.33.

```

1  function faktor_pertumbuhan (Tabel_Komponen_Penduduk, Tabel_Penduduk :
    TADOTable; ID_Komponen : Integer): Extended;
2  var
3      Komp_No : Integer;
4      Komp_Loop : Integer;
5      jumlah : Extended;
6      ValueOfKomponen : array of Extended;
7      f : array of Extended;
8  begin
9      Tabel_Komponen_Penduduk.Locate('ID_Komponen_Kependudukan', ID_Komponen, []);
10     Tabel_Penduduk.First;
11     Komp_No:=0; Komp_Loop:=0;
12     jumlah:=0;
13     SetLength(f, Tabel_Penduduk.RecordCount);
14     SetLength(ValueOfKomponen, Tabel_Penduduk.RecordCount);
15     while not Tabel_Penduduk.Eof do
16     begin
17         if Tabel_Penduduk.FieldValues['ID_Komponen_Kependudukan']=
            ID_Komponen then
18         begin
19             ValueOfKomponen[Komp_No] :=
                Tabel_Penduduk.FieldValues['Nilai'];
20             if Komp_No=0 then
21             begin
22                 Komp_No:=Komp_No+1;
23                 Komp_Loop:=Komp_Loop+1;
24             end else
25             begin
26                 f[Komp_Loop]:= (ValueOfKomponen[Komp_No]-
                    ValueOfKomponen[Komp_No-1]) / (ValueOfKomponen[Komp_No-1]);
27                 jumlah := jumlah+f[Komp_Loop];
28                 Komp_No:=Komp_No+1;
29                 Komp_Loop:=Komp_Loop+1;
30             end;
31         end;
32     end;

```

```

1  while not Tabel_Kependudukan_Populasi.Eof do
2  begin
3      ramal_anggota[Tabel_Kependudukan_Populasi.FieldName('ID_Zona
        ').AsInteger-1] :=
        Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Penduduk'] *
        Power(1+fp[0],tahun_rencana-tahun_dasar);
4      ramal_pendapatan[Tabel_Kependudukan_Populasi.FieldName('ID_Z
        ona').AsInteger-1] :=
        Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Pendapatan'] *
        Power(1+fp[1],tahun_rencana-tahun_dasar);
5      ramal_kendaraan[Tabel_Kependudukan_Populasi.FieldName('ID_Zo
        na').AsInteger-1] :=
        Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Kendaraan'] *
        Power(1+fp[2],tahun_rencana-tahun_dasar);
6      ramal_perjalanan[Tabel_Kependudukan_Populasi.FieldName('ID_Z
        ona').AsInteger-1] := Round(b0 + b1 *
        Ramal_anggota[Tabel_Kependudukan_Populasi.FieldName('ID_Zona
        ').AsInteger-1] + b2 *
        ramal_pendapatan[Tabel_Kependudukan_Populasi.FieldName('ID_Z
        ona').AsInteger-1] + b3 *
        ramal_kendaraan[Tabel_Kependudukan_Populasi.FieldName('ID_Zo
        na').AsInteger-1]);
7      Tabel_Kependudukan_Populasi.Next;
8  end;

```

Program 4.36 Program Peramalan Data Perjalanan

4.2.2.1.3. Perluasan Data Survey

Data-data hasil survey HI yang masih berupa data sampel harus diperluas menjadi jumlah data populasi, atau jumlah data yang mendekati keadaan populasi sebenarnya, dengan mengalikan data yang masih dalam jumlah sampel dengan masing-masing faktor ekspansinya.

Faktor ekspansi digunakan untuk memperluas data sampel hasil survey pada semua zona menjadi data populasi, yaitu data yang hampir menyerupai jumlah sesungguhnya. untuk mendapatkan faktor ekspansi tersebut dapat dilihat pada program 4.37, dimana data-data yang digunakan dalam rumus faktor ekspansi (program 4.37 baris 9) diperoleh dari tabel Sampel_HI (program 4.37

alah data survey perjalanan antar zona yang dilakukan, dengan daftar programnya dapat dilihat pada program 4.39.

```

1  function Ekspansi_Data_Sampel (Tabel_Sampel_HI : TADOTable; i :
    Integer): Extended;
2  var
3      a,b,c,d : Integer;
4  begin
5      a := Tabel_Sampel_HI.FieldValues['Jumlah_Seluruh_Keluarga'];
6      b := Tabel_Sampel_HI.FieldValues['Jumlah_Keluarga_Disurvey'];
7      c := Tabel_Sampel_HI.FieldValues['Jumlah_Rumah_Kosong'];
8      d := Tabel_Sampel_HI.FieldValues['Keluarga_Menolak_Diwawancara'];
9      Result := Round((a - a / b * (c + d)) / (b - c - d));
10 end;
```

Program 4.37 Unit Faktor Ekspansi

```

1  while not (Tabel_Total_Data_Keluarga.Eof) do
2  begin
3      Tabel_Kependudukan_Populasi.Append;
4      Tabel_Kependudukan_Populasi.FieldValues['ID_Zona'] :=
        Tabel_Total_Data_Keluarga.FieldValues['ID_Zona'];
5      Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Penduduk'] :=
        e[Tabel_Total_Data_Keluarga.FieldName('ID_Zona').AsInteger-1] *
        Tabel_Total_Data_Keluarga.FieldValues['Total_Anggota_Keluarga'];
6      Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Pendapatan'] :=
        e[Tabel_Total_Data_Keluarga.FieldName('ID_Zona').AsInteger-1] *
        Tabel_Total_Data_Keluarga.FieldValues['Total_Pendapatan'];
7      Tabel_Kependudukan_Populasi.FieldValues['Jumlah_Kendaraan'] :=
        e[Tabel_Total_Data_Keluarga.FieldName('ID_Zona').AsInteger-1] *
        Tabel_Total_Data_Keluarga.FieldValues['Total_Kendaraan'];
8      Tabel_Kependudukan_Populasi.Post;
9      Tabel_Total_Data_Keluarga.Next;
10 end;
```

Program 4.38 Program Perluasan Data Sampel Kependudukan

```

1  while not Tabel_Asal_Tujuan_Perjalanan.Eof do
2  begin
3      Tabel_Perjalanan_Populasi.Append;
4      Tabel_Perjalanan_Populasi.FieldValues['Zona_Asal'] :=
```


4. 2. 2. 1. 4. Peramalan Perjalanan Antar Zona

Dalam peramalan distribusi perjalanan atau peramalan perjalanan antar zona digunakan metode *furness*, dalam metode ini tiap data perjalanan yang ada dikalikan dengan faktor pertumbuhan bangkitan tiap zona dan faktor pertumbuhan tarikan tiap zona secara bergantian untuk setiap perulangan atau iterasi yang terjadi. Pada perulangan pertama, data tiap perjalanan antar zona dikalikan dengan faktor pertumbuhan bangkitan tiap zona terlebih dahulu (baris 6 sampai 9), kemudian perulangan kedua dikalikan dengan faktor pertumbuhan tarikan tiap zona (baris 10 sampai 14), perulangan ketiga dikalikan dengan faktor pertumbuhan bangkitan tiap zona, dan seterusnya (baris 6 sampai baris 14) sampai total bangkitan dan total tarikan hasil peramalan sama dengan total bangkitan dan total tarikan hasil ramalan dengan persamaan regresi yang didapatkan pada tahapan peramalan perjalanan sebelumnya untuk semua zona (baris 1). Dengan asumsi awal dikarenakan perbandingan yang ada kecil kemungkinan benar-benar sama dengan satu, maka diasumsikan hasil perbandingan yang sama cukup mendekati nilai 1, yaitu dengan nilai antara 0.999 dengan 1.001. selengkapnya dapat dilihat pada program 4.40.

4. 2. 2. 2. Proses Pencarian Rute Tercepat Antar Zona

Pada proses ini akan dicari rute-rute tercepat antara satu zona dengan zona lainnya. Dimulai dari pergerakan antara zona yang mempunyai jumlah terbesar sampai dengan jumlah terkecil, yang diambil dari *view* tabel asal tujuan perjalanan. Pada *view* ini perjalanan yang ada pada tabel asal tujuan perjalanan diurutkan dari perjalanan dengan nilai paling besar sampai pada perjalanan dengan nilai paling kecil (program 4.41).

panjang dari rute alternatif tersebut (baris 15 dan 16). dari proses ini dapat dilihat ada program 4.42.

```

1  while not (((0.999 <
    (ramal_perjalanan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-
1]/perjalanan_bangkitan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-1]))
and (1.001 >
    (ramal_perjalanan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-
1]/perjalanan_bangkitan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-1]))
and (((0.999 <
    (ramal_perjalanan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-
1]/perjalanan_tarikan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-1]))
and (1.001 >
    (ramal_perjalanan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-
1]/perjalanan_tarikan[Tabel_Zona.FieldByName('ID_Zona').AsInteger-1]))))
do
2  begin
3      for awal_zona := 1 to Tabel_Zona.RecordCount do
4          for akhir_zona := 1 to Tabel_Zona.RecordCount do
5              begin
6                  if looping mod 2 = 1 then
7                      begin
8                          pertumbuhan_zona[awal_zona-1]:=(ramal_perjalanan[awal_zona-
9                          1] / perjalanan_bangkitan[awal_zona-1]);
10                         ramal_zona[awal_zona-1,akhir_zona-1]:=
11                         (perjalanan_zona[awal_zona-1,akhir_zona-
12                         1]*pertumbuhan_zona[awal_zona-1]);
13                     end else if looping mod 2 = 0 then
14                         begin
15                             pertumbuhan_zona[akhir_zona-1]:=
16                             ramal_perjalanan[akhir_zona-1] /
17                             perjalanan_tarikan[akhir_zona-1];
18                             ramal_zona[awal_zona-1,akhir_zona-1]:=
19                             (perjalanan_zona[awal_zona-1,akhir_zona-
20                             1]*pertumbuhan_zona[akhir_zona-1]);
21                         end;
22                     total_bangkitan_ramal[awal_zona-1] :=
23                     total_bangkitan_ramal[awal_zona-1] + ramal_zona[awal_zona-
24                     1,akhir_zona-1];
25                     total_tarikan_ramal[akhir_zona-1] :=
26                     total_tarikan_ramal[akhir_zona-1]+ ramal_zona[awal_zona-
27                     1,akhir_zona-1];
28                 end;
29             end;
30         end;
31     end;
32 end;

```



```

1  DataModuleSPK.View_Urutan_Jumlah_Perjalanan.First;
2  i := 0;
3  while not DataModuleSPK.View_Urutan_Jumlah_Perjalanan.Eof do
4  begin
5      if
        DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Asal']
        ] <>
        DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Tujuan']
        n'] then
6      begin
7          Node_Zona_Asal :=
            DataModuleSPK.Tabel_Node.Lookup('ID_Zona;Node_Pusat',VarArrayOf
            ([DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Asal'],True]),'ID_Node');
8          Node_Zona_Tujuan :=
            DataModuleSPK.Tabel_Node.Lookup('ID_Zona;Node_Pusat',VarArrayOf
            ([DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Tujuan'],True]),'ID_Node');
9          Alternative[i].Zona_Asal :=
            DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Asal'];
10         Alternative[i].Zona_Tujuan :=
            DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Zona_Tujuan'];
11         Alternative[i].Rute := Dijkstra(Node_Zona_Asal,
            Node_Zona_Tujuan, DataModuleSPK.Tabel_Node,
            DataModuleSPK.Tabel_Node_Link, DataModuleSPK.View_Link_Jalan,
            False, Weight, TrayekVisited);
12         for j := 0 to (Length(Alternative[i].Rute) div 5 - 1) do
13         begin
14             DataModuleSPK.View_Link_Jalan.Locate('Nama_Link',
                Copy(Alternative[i].Rute, j*5+1, 9), []);
15             Alternative[i].Waktu :=
                DataModuleSPK.View_Link_Jalan.FieldValues['Panjang'] /
                (DataModuleSPK.View_Link_Jalan.FieldValues['Kecepatan'] *
                1000 / 60);
16             Alternative[i].Panjang :=
                DataModuleSPK.View_Link_Jalan.FieldValues['Panjang'];
17             Weight[DataModuleSPK.View_Link_Jalan.RecNo-1] :=
                Weight[DataModuleSPK.View_Link_Jalan.RecNo-1] +
                DataModuleSPK.View_Urutan_Jumlah_Perjalanan.FieldValues['Jumlah_Perjalanan'];
18         end;
19         Inc(i);
20     end;
21     DataModuleSPK.View_Urutan_Jumlah_Perjalanan.Next;
22 end;

```


dan arah kepulangan dari rute trayek angkutan kota tersebut. Dengan nilai satu berarti bahwa rute trayek berjalan untuk arah keberangkatan, dan nilai 2 untuk arah kepulangan.

Maka langkah awal yang dilakukan untuk tiap-tiap rute trayek angkutan kota dan jenis arah rutenya adalah menetapkan semua *link* yang ada tidak pernah dilewati oleh rute trayek angkutan kota (program 4.43 baris 1). Dan melakukan inisialisasi awal untuk rute lama dan rute baru (program 4.43 baris 2 sampai 5), serta memberi nilai variabel *checked* sama dengan *false* (program 4.43 baris 6). Untuk semua *link* yang dilalui oleh himpitan antara rute trayek dengan alternatif rute tercepat diberi tanda bahwa *link* tersebut telah dilewati oleh rute trayek tersebut, dan begitu juga untuk *link* arah sebaliknya (program 4.43 baris 7 sampai 15). Daftar program selengkapnya dapat dilihat pada program 4.43.

Setelah semua *link* yang dilewati rute angkutan kota untuk satu arah keberangkatan diberi tanda, kemudian dimulai lagi dari awal memeriksa *link-link* yang dilewati rute angkutan kota tersebut satu persatu. Dengan langkah awal memberi nilai untuk rute lama pada *link* yang pertama kali dilalui. Dapat dilihat pada program 4.44. Kemudian untuk semua *link* rute yang ada pada satu jaringan trayek dan satu arah keberangkatan, dilihat apakah pada *link* rute tersebut ada pada *link* rute tercepat antar zona (baris 8). Bila rute lama trayek angkutan kota tersebut ada pada rute alternatif tercepat antar zona dan tanda *checked* sudah di-true-kan (baris10), maka *node* awal pada jalan atau *link* rute tersebut dijadikan titik akhir atau titik tujuan (baris 12) dalam pencarian rute tercepat dengan menggunakan algoritma *dijkstra* (baris 13), dan kemudian tanda *checked* di-false-kan (baris 27). Daftar program selengkapnya dapat dilihat pada program

false, maka rute baru mempunyai rute yang sama dengan rute lama
 am 4.46 baris 32 sampai 38).

```

for k := 0 to LinkCount-1 do TrayekVisited[k] := False;
Trayek_Lama[i][j].Rute :=
Copy(DataModuleSPK.View_Link_Jalan.Lookup('ID_Link',
DataModuleSPK.Tabel_Rute.FieldByName('ID_Link').AsInteger, 'Nama_Link'),
1, 4);
Trayek_Lama[i][j].Waktu := 0;
Trayek_Baru[i][j].Rute :=
Copy(DataModuleSPK.View_Link_Jalan.Lookup('ID_Link',
DataModuleSPK.Tabel_Rute.FieldByName('ID_Link').AsInteger, 'Nama_Link'),
1, 4);
Trayek_Baru[i][j].Waktu := 0;
Checked := False;
while not DataModuleSPK.Tabel_Rute.Eof do
begin
  DataModuleSPK.View_Link_Jalan.Locate('ID_Link',
  DataModuleSPK.Tabel_Rute.FieldByName('ID_Link').AsInteger, []);
  if Weight[DataModuleSPK.View_Link_Jalan.RecNo-1] <> 0 then
  begin
    TrayekVisited[DataModuleSPK.View_Link_Jalan.RecNo-1] := True;
    if
      DataModuleSPK.View_Link_Jalan.Locate('ID_Node_Awal;ID_Node_Akhir'
      ,
      VarArrayOf([DataModuleSPK.View_Link_Jalan.FieldValues['ID_Node_Akhir'],
      DataModuleSPK.View_Link_Jalan.FieldValues['ID_Node_Awal']])
      , []) then
      TrayekVisited[DataModuleSPK.View_Link_Jalan.RecNo-1] := True;
  end;
  DataModuleSPK.Tabel_Rute.Next;
end;

```

Program 4.43 Program Mencari rute baru (1)

```

DataModuleSPK.Tabel_Rute.First;
while not DataModuleSPK.Tabel_Rute.Eof do
begin

```

```

8  if Weight[DataModuleSPK.View_Link_Jalan.RecNo-1] <> 0 then
9  begin
10     if Checked then
11     begin
12         Node_Zona_Tujuan :=
13         DataModuleSPK.View_Link_Jalan.FieldValues['ID_Node_Awal'];
14         OldRecPos := DataModuleSPK.View_Link_Jalan.GetBookmark;
15         Route := Dijkstra(Node_Zona_Asal, Node_Zona_Tujuan,
16         DataModuleSPK.Tabel_Node, DataModuleSPK.Tabel_Node_Link,
17         DataModuleSPK.View_Link_Jalan, True, Weight, TrayekVisited);
18         if not (Route = '') then
19         begin
20             for k := 0 to (Length(Route) div 5 - 1) do
21             begin
22                 DataModuleSPK.View_Link_Jalan.Locate('Nama_Link',
23                 Copy(Route, k*5+1, 9), []);
24                 Trayek_Baru[i][j].Waktu := Trayek_Baru[i][j].Waktu +
25                 DataModuleSPK.View_Link_Jalan.FieldValues['Panjang'] /
26                 (DataModuleSPK.View_Link_Jalan.FieldValues['Kecepatan'] *
27                 1000 / 60);
28                 Trayek_Baru[i][j].Panjang := Trayek_Baru[i][j].Panjang +
29                 DataModuleSPK.View_Link_Jalan.FieldValues['Panjang'];
30                 TrayekVisited[DataModuleSPK.View_Link_Jalan.RecNo-1] :=
31                 True;
32                 if
33                 DataModuleSPK.View_Link_Jalan.Locate('ID_Node_Awal;ID_Node
34                 Akhir',
35                 VarArrayOf([DataModuleSPK.View_Link_Jalan.FieldValues['ID_
36                 Node_Akhir'],
37                 DataModuleSPK.View_Link_Jalan.FieldValues['ID_Node_Awal']])
38                 ), []) then
39                     TrayekVisited[DataModuleSPK.View_Link_Jalan.RecNo-1] :=
40                     True;
41                 end;
42                 Trayek_Baru[i][j].Rute := Copy(Trayek_Baru[i][j].Rute, 1,
43                 Length(Trayek_Baru[i][j].Rute)-4) + Route;
44                 Checked := False;
45             end;
46             DataModuleSPK.View_Link_Jalan.GotoBookmark(OldRecPos);
47             DataModuleSPK.View_Link_Jalan.FreeBookmark(OldRecPos);
48         end;
49     end;
50 end;

```

Program 4.45 Program mencari rute baru (3)

```

32  if (not Checked) then
33  begin
34      Trayek_Baru[i][j].Rute := Trayek_Baru[i][j].Rute +
35      Copv(DataModuleSPK.View_Link_Jalan.FieldValues['Nama_Link'], 5,

```


Dan akhirnya pada hasil akhir ditemukan rute baru trayek angkutan kota dengan titik awal dan titik akhirnya sama dengan rute lama trayek angkutan kota, hanya saja pada rute di dalamnya, terdapat beberapa kemungkinan ada *link-link* yang berubah.

4. 2. 2. 4. Proses Evaluasi RWP dan RBP Rute Baru Terhadap kendaraan Pribadi

Evaluasi RWP dan RBP dilakukan pada rute lama dan rute baru trayek angkutan kota. Dimana RWP dan RBP adalah rasio perbandingan waktu dan biaya antara penggunaan trayek dan penggunaan kendaraan pribadi. RWP dan RBP ini dihitung untuk tiap titik perpotongan pusat zona yang dilalui oleh rute trayek tersebut. Dimana perhitungan tersebut dimulai dengan melakukan perhitungan terhadap waktu dan panjang dari rute angkutan kota antara 2 pusat zona. Perhitungan waktu dan panjang dilakukan dengan melakukan penambahan waktu (baris 12) dan panjang (baris 13) dari tiap *link* yang dilalui, dan terus dilakukan bersamaan dengan pengecekan *node* yang dilalui oleh rute trayek, bila *node* yang dilalui rute tersebut adalah *node* pusat (baris 14), maka proses pengecekan dan penambahan nilai waktu dan panjang rute trayek dihentikan sehingga didapatkan zona tujuan yang merupakan zona dari *node* pusat tersebut (baris 20), dan zona asal adalah zona tempat *node* awal rute trayek angkutan kota (baris 5 dan baris 21). Pada pengecekan *node* atau perulangan selanjutnya zona tujuan akan dijadikan sebagai zona asal yang baru (baris 26), sedangkan zona tujuan yang baru adalah zona tempat *node* pusat selanjutnya ditemukan. Kemudian dilanjutkan dengan mencari waktu dan panjang rute yang dilalui oleh kendaraan pribadi. Waktu dan panjang rute kendaraan pribadi diasumsikan

sama dengan zona asal dan tujuan rute trayek, maka pengecekan akan terus dilakukan, sampai ditemukan zona asal dan tujuan yang sama dengan rute trayek (baris 17). Setelah mendapatkan waktu dan panjang rute dengan menggunakan trayek angkutan kota dan kendaraan pribadi, dilanjutkan dengan mencari nilai RWP, yang merupakan perbandingan antara waktu perjalanan rute trayek dibagi waktu perjalanan rute kendaraan pribadi (baris 22). Sedangkan nilai RBP didapat dari perbandingan biaya menggunakan angkutan, yang mempunyai biaya yang sama baik untuk jarak dekat maupun jauh, yaitu Rp 2000 dibagi dengan biaya menggunakan kendaraan pribadi, yang diperoleh dengan mengalikan biaya per meternya dengan panjang rute kendaraan pribadi (baris 23). Kemudian hasilnya disimpan dalam suatu tabel sementara, yaitu tabel evaluasi trayek untuk menyimpan evaluasi RWP dan RBP trayek rute baru, dan tabel evaluasi trayek lama untuk menyimpan evaluasi RWP dan RBP trayek rute lama. Dimana kedua tabel ini akan terus berganti nilainya sesuai dengan hasil proses yang dijalankan. Kemudian perhitungan tersebut terus dilakukan sampai rute trayek angkutan kota habis dan seluruh trayek baik untuk arah keberangkatan dan kepulangan telah diperiksa dan dihitung nilai RWP dan RBP-nya. Dan pada dasarnya seluruh proses untuk mencari nilai RWP dan RBP tersebut untuk rute lama dan rute baru adalah sama. Selanjutnya daftar program selengkapnya untuk mencari evaluasi nilai RWP dan RBP dapat dilihat pada program 4.49 untuk evaluasi rute baru, dan program 4.50 untuk evaluasi rute lama.

4. 2. 2. 5. Proses Persentase Dan Pangsa Pasar Penggunaan Angkutan

Mencari persentase dan pangsa pasar trayek angkutan kota dilakukan untuk melihat kinerja angkutan kota baik setelah menggunakan rute baru maupun

persentase penggunaan angkutan kota untuk rute baru, maka variabel-variabel dari rute lama diregresikan sehingga akan diperoleh persamaan *regresi* (baris 4). Setelah persamaan *regresi* tersebut didapat maka persentase penggunaan angkutan kota untuk rute baru dapat dicari (baris 12 sampai 18). Kemudian hasilnya dimasukkan ke dalam tabel evaluasi trayek. Setelah diperoleh persentase penggunaan angkutan kota, maka pangsa pasar trayek angkutan kota juga dapat dicari dengan mengalikan persentase penggunaan angkutan kota dengan jumlah total perjalanan (baris 20). Daftar program selengkapnya dapat dilihat pada program 4.51.

2. 2. 6. Unit Dijkstra

Selanjutnya akan dijelaskan sebuah unit yang digunakan dalam proses pencarian rute tercepat antar zona dan pencarian rute tercepat antara dua buah titik potongan rute lama dengan rute tercepat antar zona yang telah dijelaskan pada sub-bab 4.2.2.2 dan 4.2.2.3, dan unit tersebut adalah unit dijkstra. Unit dijkstra digunakan untuk mencari rute tercepat antara dua buah titik pusat zona. Dalam unit ini terdapat beberapa fungsi yang digunakan, yang akan dijelaskan pada sub-bab berikut.

2. 2. 6. 1. Fungsi GetRecNo

Fungsi GetRecNo berfungsi untuk mencatat dan menyimpan nomor


```

1  while not DataModuleSPK.Tabel_Evaluasi_Trayek.Eof do
    DataModuleSPK.Tabel_Evaluasi_Trayek.Delete;
2  for i := 0 to TrayekCount-1 do
3      for j := 0 to 1 do
4          begin
5              Zona := DataModuleSPK.Tabel_Node.Lookup('Nama_Node',
                Copy(Trayek_Baru[i][j].Rute, 1, 4), 'ID_Zona');
6              RWP := 0;
7              Panjang_Rute := 0;
8              for k := 0 to (Length(Trayek_Baru[i][j].Rute) div 5 - 1)
                do
9                  begin
10                     DataModuleSPK.View_Link_Jalan.Locate('Nama_Link',
                            Copy(Trayek_Baru[i][j].Rute, k*5+1, 9), []);
11                     DataModuleSPK.Tabel_Node.Locate('Nama_Node',
                            Copy(Trayek_Baru[i][j].Rute, (k+1)*5+1, 4), []);
12                     RWP := RWP +
                            DataModuleSPK.View_Link_Jalan.FieldValues['Panjang']
                            /
                            (DataModuleSPK.View_Link_Jalan.FieldValues['Kecepatan
                            ']* 1000 / 60);
13                     Panjang_Rute := Panjang_Rute +
                            DataModuleSPK.View_Link_Jalan.FieldValues['Panjang'];
14                     if DataModuleSPK.Tabel_Node.FieldValues['Node_Pusat']
                            = True then
15                         begin
16                             l := 0;
17                             while (Alternative[l].Zona_Asal <> Zona) or
                                    (Alternative[l].Zona_Tujuan <>
                                    DataModuleSPK.Tabel_Node.FieldValues['ID_Zona'])
                                    do Inc(l);
18                             DataModuleSPK.Tabel_Evaluasi_Trayek.Append;
19                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['T
                                    rayek'] := i+1;
20                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['A
                                    khir_Zona'] :=
                                    DataModuleSPK.Tabel_Node.FieldValues['ID_Zona'];
21                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['A
                                    wal_Zona'] := Zona;
22                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['R
                                    WP'] := RWP / (Alternative[l].Waktu);
23                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['R
                                    BP'] :=
                                    DataModuleSPK.Tabel_Trayek.Lookup('ID_Trayek',
                                    i+1, 'Tarif') / (0.550 * Alternative[l].Panjang);
24                             DataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['P
                                    anjang_Rute'] := Panjang_Rute;
25                             DataModuleSPK.Tabel_Evaluasi_Trayek.Post;
26                             Zona :=
                                    DataModuleSPK.Tabel_Node.FieldValues['ID_Zona'];

```

```

1  while not DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.Eof do
    DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.Delete;
2  for i := 0 to TrayekCount-1 do
3      for j := 0 to 1 do
4          begin
5              Zona_Lama := DataModuleSPK.Tabel_Node.Lookup('Nama_Node',
                Copy(Trayek_Lama[i][j].Rute, 1, 4), 'ID_Zona');
6              RWP_Lama := 0;
7              Panjang_Rute_Lama := 0;
8              for k := 0 to (Length(Trayek_Lama[i][j].Rute) div 5 - 1) do
9                  begin
10                     DataModuleSPK.view_Link_Jalan.Locate('Nama_Link',
                            Copy(Trayek_Lama[i][j].Rute, k*5+1, 9), []);
11                     DataModuleSPK.Tabel_Node.Locate('Nama_Node',
                            Copy(Trayek_Lama[i][j].Rute, (k+1)*5+1, 4), []);
12                     RWP_Lama := RWP_Lama +
                            DataModuleSPK.view_Link_Jalan.FieldValues['Panjang'] /
                            (DataModuleSPK.view_Link_Jalan.FieldValues['Kecepatan'] *
                            1000 / 60);
13                     Panjang_Rute_Lama := Panjang_Rute_Lama +
                            DataModuleSPK.view_Link_Jalan.FieldValues['Panjang'];
14                     if DataModuleSPK.Tabel_Node.FieldValues['Node_Pusat'] =
                            True then
15                         begin
16                             l := 0;
17                             while (Alternative[l].Zona_Asal <> Zona_Lama) or
                                    (Alternative[l].Zona_Tujuan <>
                                    DataModuleSPK.Tabel_Node.FieldValues['ID_Zona']) do
                                Inc(l);
18                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.Append;
19                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['T
                                    rayek'] := i+1;
20                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['A
                                    khir_Zona'] :=
                                    DataModuleSPK.Tabel_Node.FieldValues['ID_Zona'];
21                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['A
                                    wal_Zona'] := Zona_Lama;
22                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['R
                                    WP'] := RWP_Lama / (Alternative[l].Waktu);
23                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['R
                                    BP'] := DataModuleSPK.Tabel_Trayek.Lookup('ID_Trayek',
                                    i+1, 'Tarif') / (0.550 * Alternative[l].Panjang);
24                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['P
                                    anjang_Rute'] := Panjang_Rute_Lama;
25                             DataModuleSPK.Tabel_Asal_Tujuan_Perjalanan.Locate('Zona
                                    Asal;Zona_Tujuan',
                                    VarArrayOf([Zona_Lama, DataModuleSPK.Tabel_Node.FieldVal
                                    ues['ID_Zona']]), []);
26                             DataModuleSPK.Tabel_Evaluasi_Trayek_Lama.FieldValues['P
                                    rosentase_Penggunaan'] :=
                                    (DataModuleSPK.Tabel_Asal_Tujuan_Perjalanan.FieldByName
                                    ('Jumlah_Perjalanan_Angkutan')).AsInteger/DataModuleSPK.
                                    Tabel_Asal_Tujuan_Perjalanan.FieldByName('Jumlah_Perjal

```

```

1  SetLength(Pangsa_Pasar,DataModuleSPK.Tabel_Trayek.RecordCount);
2  for Trayek := 1 to DataModuleSPK.Tabel_Trayek.RecordCount do
3  begin
4      Regresi_Linier_Prosentase(bp0,bp1,bp2,bp3,bp4,bp5,bp6,'View_Gabu
      ngan_Evaluasi','Jumlah_Penduduk','Jumlah_Pendapatan','Jumlah_Ken
      daraan','RWP','RBP','Panjang_Rute','Prosentase_Penggunaan',DataM
      oduleSPK.ADOQuery1,Trayek);
5      DataModuleSPK.Tabel_Evaluasi_Trayek.Filtered := False;
6      DataModuleSPK.Tabel_Evaluasi_Trayek.Filter := 'Trayek =
      '+IntToStr(Trayek);
7      DataModuleSPK.Tabel_Evaluasi_Trayek.Filtered := True;
8      DataModuleSPK.Tabel_Evaluasi_Trayek.First;
9      for AsalTujuan := 1 to
      (DataModuleSPK.Tabel_Evaluasi_Trayek.RecordCount) do
10     begin
11         DataModuleSPK.Tabel_Kependudukan_Populasi.Locate('ID_Zona',D
      ataModuleSPK.Tabel_Evaluasi_Trayek.FieldValues['Awal_Zona'],
      []);
12         Prosentase_Trayek :=  bp0 +
13         (bp1 *
      DataModuleSPK.Tabel_Kependudukan_Populasi.FieldByName('Jumla
      h_Penduduk').AsInteger) +
14         (bp2 *
      DataModuleSPK.Tabel_Kependudukan_Populasi.FieldByName('Jumla
      h_Pendapatan').AsInteger) +
15         (bp3 *
      DataModuleSPK.Tabel_Kependudukan_Populasi.FieldByName('Jumla
      h_Kendaraan').AsInteger) +
16         (bp4 *
      log10(DataModuleSPK.Tabel_Evaluasi_Trayek.FieldByName('RWP')
      .AsInteger)) +
17         (bp5 *
      log10(DataModuleSPK.Tabel_Evaluasi_Trayek.FieldByName('RBP')
      .AsInteger)) +
18         (bp6 *
      DataModuleSPK.Tabel_Evaluasi_Trayek.FieldByName('Panjang_Rut
      e').AsInteger);
19         DataModuleSPK.Tabel_Perjalanan_Populasi.Locate('Zona_Asal;Zo
      na_Tujuan',VarArrayOf([DataModuleSPK.Tabel_Evaluasi_Trayek.F
      ieldValues['Awal_Zona'],DataModuleSPK.Tabel_Evaluasi_Trayek.
      FieldValues['Akhir_Zona']]),[]);
20         Pangsa_Pasar_Angkot := (Prosentase_Trayek/100) *
      DataModuleSPK.Tabel_Perjalanan_Populasi.FieldByName('Perjala
      nan_Populasi').AsInteger;
21         DataModuleSPK.Update_Tabel.Parameters[0].Value :=
      Prosentase_Trayek;
22         DataModuleSPK.Update_Tabel.Parameters[1].Value :=
      Pangsa_Pasar_Angkot;
23         DataModuleSPK.Update_Tabel.Parameters[2].Value := Trayek;
24         DataModuleSPK.Update_Tabel.Parameters[3].Value :=
      DataModuleSPK.Tabel_Evaluasi_Trayek.FieldByName('Awal_Zona')
      .AsInteger;
25         DataModuleSPK.Update_Tabel.Parameters[4].Value :=

```


2.2.6.2. Fungsi DirectPathCost1

Fungsi `DirectPathCost1` digunakan untuk memberikan biaya atau beban ke *link* yang ditunjuk, dengan *node* awalnya merupakan titik awal dalam proses pemilihan rute. Biaya tersebut diberikan bila terdapat *node* akhir yang mempunyai hubungan *link* dengan *node* awal (program 4.53 baris 9 sampai 15), bila antara *node* awal dan *node* akhir tidak ada hubungan maka biaya yang diberikan adalah *infinity* (program 4.53 baris 18). Biaya yang diberikan itu merupakan biaya waktu yang tercepat dari semua rute alternatif yang ada dari titik awal (*start*) ke titik tujuan (*goal*). Waktu diperoleh dengan membagi nilai panjang *link* dengan kecepatan di *link* tersebut. Untuk fungsi tersebut dapat dilihat pada program 4.53.

```

1 function GetRecNo(Node : Integer): Integer;
2 var OldRecNo : TBookmark;
3 begin
4   OldRecNo := Data_Node.GetBookmark; // Simpan posisi record
5   if Data_Node.Locate('ID_Node', Node, []) then
6     Result := Data_Node.RecNo
7   else
8     Result := 0;
9   Data_Node.GotoBookmark(OldRecNo); // Kembalikan posisi record
10  Data_Node.FreeBookmark(OldRecNo);
11 end;
```

Program 4.52 Fungsi GetRecNo

2.2.6.3. Fungsi DirectPathCost2

Fungsi `DirectPathCost2` hampir sama dengan `DirectPathCost1`, perbedaannya hanya pada tujuan dan kegunaan dari fungsi ini. Fungsi `DirectPathCost2` digunakan untuk memberikan biaya pada *link* yang digunakan untuk pemilihan rute trayek. Sehingga pada fungsi ini terdapat

```

1 function DirectPathCost1(Start, Goal : Integer): Real;
2 var
3   OldRecNo : TBookmark;
4 begin
5   OldRecNo := Data_Node.GetBookmark; // Simpan posisi record
6   Data_Node.Locate('ID_Node', Start, []);
7   while not Data_Link.Eof do
8   begin
9     if Data_Link.FieldValues['ID_Node_Akhir'] = Goal then
10    begin
11      Result:=(Data_Link.FieldValues['Panjang']/
12      Data_Link.FieldValues ['Kecepatan'] );
13      Data_Node.GotoBookmark(OldRecNo); // Kembalikan posisi record
14      Data_Node.FreeBookmark(OldRecNo);
15      Exit;
16    end;
17    Data_Link.Next;
18  end;
19  Result := Infinite;
20  Data_Node.GotoBookmark(OldRecNo); // Kembalikan posisi record
21  Data_Node.FreeBookmark(OldRecNo);
22 end;

```

Program 4.53 Fungsi DirectPathCost1

```

1 function DirectPathCost2(Start, Goal : Integer): Integer;
2 var OldRecNo : TBookmark;
3 begin
4   OldRecNo := Data_Node.GetBookmark; Data_Node.Locate('ID_Node',
5   Start, []);
6   if Data_Link.Locate('ID_Node_Awal;ID_Node_Akhir',
7   VarArrayOf([Start,Goal]), []) then
8   begin
9     Data_All_Link.Locate('ID_Link',
10     Data_Link.FieldValues['ID_Link'], []);
11     if not TrayekVisited[Data_All_Link.RecNo-1] then
12     begin
13       Data_Link.First;
14       while not Data_Link.Eof do
15       begin
16         if Data_Link.FieldValues['ID_Node_Akhir'] = Goal then
17         begin
18           Result := (Data_Link.FieldValues['Panjang'] /
19           Data_Link.FieldValues['Kecepatan']);
20           Data_Node.GotoBookmark(OldRecNo);
21           Data_Node.FreeBookmark(OldRecNo);
22           Exit;
23         end;
24       end;
25     end;
26   end;

```

4.2.2.6.4. Fungsi Dijkstra

Fungsi *dijkstra* digunakan untuk mendapatkan rute antara titik awal dengan titik akhir dengan biaya paling optimal. Fungsi ini menggunakan ketiga fungsi yang telah dijelaskan pada sub bab sebelumnya, yaitu sub-bab 4.2.2.6.1 sampai 4.2.2.6.3. Kegunaan fungsi *dijkstra* dalam pencarian rute baru trayek angkutan kota adalah untuk mencari rute tercepat untuk semua zona, dan untuk mencari rute tercepat antara dua titik perpisahan antara himpitan jalur trayek lama dengan jalur alternatif rute tercepat semua zona yang telah diperoleh sebelumnya. Untuk mendapatkan rute tercepat antar zona digunakan fungsi *DirectPathCost1* yang digunakan untuk mendapatkan biaya tiap link yang ditunjuk. Sedangkan untuk mendapatkan rute tercepat antara dua titik perpisahan himpitan jalur trayek lama dan alternatif jalur digunakan fungsi *DirectPathCost2*.

Jalannya proses *dijkstra* ini dimulai dengan mencari *node-node* yang berhubungan dengan *start*, yang merupakan titik awal dari rute yang akan dicari. Semua *node* yang ada diperiksa satu persatu hingga ditemukan *node-node* yang mempunyai hubungan *link* dengan titik awal (*start*) tersebut. Bila yang memanggil fungsi *dijkstra* adalah proses mencari rute tercepat antar zona, untuk memeriksanya digunakan fungsi *DirectPathCost1* (program 4.55 baris 7 dan 8), seperti yang telah dijelaskan pada sub bab 4.2.2.1.2. Sedangkan bila yang memanggil adalah proses untuk mendapatkan rute tercepat antara 2 titik perpisahan himpitan rute trayek dengan rute tercepat antar zona yang telah didapatkan sebelumnya, untuk memeriksanya digunakan fungsi *DirectPathCost2* (program 4.55 baris 5 dan 6). Bila *node* yang ditunjuk

paris 16 sampai 20). untuk tahapan proses tersebut dapat dilihat pada program 4.57.

```

1  MinCost := Infinite;
2  MinNode := 0;
3  Data_Node.First;
4  while not Data_Node.Eof do
5  begin
6      if not Visited[Data_Node.RecNo-1] then
7          if TotalCost[Data_Node.RecNo-1] < MinCost then
8              begin
9                  MinNode := Data_Node.FieldValues['ID_Node'];
10                 MinCost := TotalCost[Data_Node.RecNo-1];
11             end;
12         Data_Node.Next;
13     end;
14     Visited[GetRecNo(MinNode)-1] := True;

```

Program 4.56 Fungsi Dijkstra (2)

```

1  while not Data_Node.Eof do
2  begin
3      if not Visited[Data_Node.RecNo-1] then
4          begin
5              if TotalCost[GetRecNo(MinNode)-1] = Infinite then
6                  NewValue := Infinite
7              else
8                  begin
9                      if UsedWeight then
10                         NewValue := DirectPathCost2(MinNode,
11                                                         Data_Node.FieldValues['ID_Node'])
12                     else
13                         NewValue := DirectPathCost1(MinNode,
14                                                         Data_Node.FieldValues['ID_Node']);
15                     if NewValue <> Infinite then
16                         NewValue := NewValue + TotalCost[GetRecNo(MinNode)-1];
17                     end;
18                     if NewValue < TotalCost[Data_Node.RecNo-1] then
19                         begin
20                             TotalCost[Data_Node.RecNo-1] := NewValue;
21                             Node[Data_Node.RecNo-1] := MinNode;
22                         end;
23                     end;
24                 Data_Node.Next;
25             end;

```

titik terakhir adalah *infiniy*, maka tidak ditemukan rute antara titik awal dan titik akhir. Seperti dapat dilihat pada 4.58.

```

1  if TotalCost[GetRecNo(Goal)-1] = Infinite then
2  begin
3      Result := '';
4      Exit;
5  end;
```

Program 4.58 Fungsi Dijkstra (4)

Selanjutnya hasil rute tersebut ditulis dalam bentuk *string*, dimana dalam penulisannya dimulai dari titik terakhir, sampai dengan titik awal, sehingga terbentuk suatu rute, daftar programnya dapat dilihat pada program 4.59.

```

1  Route := Trim(Data_Node.Lookup('ID_Node', Goal, 'Nama_Node'));
2  CurrentNode := Goal;
3  while CurrentNode <> Start do
4  begin
5      NextNode := Node[GetRecNo(CurrentNode)-1];
6      Route := Trim(Data_Node.Lookup('ID_Node', NextNode, 'Nama_Node')) +
7      '-' + Route;
8      CurrentNode := NextNode;
9  end;
10 Result := Route;
```

Program 4.59 Fungsi Dijkstra (5)

4.3. PEMBUATAN ANTARMUKA APLIKASI

Antarmuka dalam program perencanaan trayek angkutan kota terdiri dari 3 bagian utama, yaitu antarmuka untuk halaman utama, antarmuka untuk proses masukan data, antarmuka untuk proses perencanaan trayek angkutan kota. Dalam pembuatan antarmuka tersebut digunakan *tools* dari Borland Delphi 7. Selanjutnya masing-masing antarmuka tersebut akan dijelaskan pada sub-bab

Angkutan dan Laporan. Masing-masing menu tersebut akan dijelaskan pada sub-bab berikut. Selanjutnya tampilan pertama kali (*home*) dapat dilihat pada gambar 4.33.

4.3.2. Antarmuka Untuk Masukan Data (Tampilan Masukan)

Masukan data mempunyai 4 jenis tampilan, yaitu tampilan untuk masukan data kependudukan, masukan trayek, masukan survey HI, dan masukan data kewilayahan.



Gambar 4.33 Tampilan Utama (*Home*)

4.3.2.1. Tampilan Masukan Data Kependudukan

Tampilan Masukan Data Kependudukan berfungsi untuk memasukkan data kependudukan untuk periode waktu beberapa tahun, yang digunakan untuk melakukan peramalan data kependudukan pada suatu tahun rencana. Terdapat 3 jenis tampilan untuk data kependudukan, yaitu:

- Tampilan untuk masukan data jumlah penduduk, merupakan antarmuka dari masukan data jumlah penduduk. Tampilan ini dapat dilihat pada gambar 4.34.

Data Jumlah Penduduk

Jumlah Penduduk

Tahun: 2001

Jumlah Penduduk: 107123 orang

Tahun	Jumlah Penduduk
2001	107123
1999	10802
2000	10885

Selesai

Gambar 4.34 Tampilan Untuk Masukan Jumlah Penduduk

Data Jumlah Pendapatan

Jumlah Pendapatan Per Orang

Tahun: 2000

Jumlah Pendapatan (Rp.): 6225089

Tahun	Jumlah Pendapatan (Rp.)
2000	6225089
2001	7149020
2002	8055012
2003	9041024

Selesai

Gambar 4.35 Tampilan Untuk Masukan Jumlah Pendapatan

Data Jumlah Kendaraan

Jumlah Kendaraan

Tahun: 2002

Jumlah Kendaraan: 23312 kendaraan

Tahun	Jumlah Kendaraan
2002	23312
2003	26562
2004	30641

Selesai

Gambar 4.36 Tampilan Untuk Masukan Jumlah Kendaraan

The screenshot shows a software window titled "Data Trayek" with a sub-header "Masukan Data Trayek". It contains four input fields, each with a navigation bar below it:

- "Nama Trayek": A text input field.
- "Tarif Kendaraan": A numeric input field with the value "2000".
- "No Rute": A numeric input field with the value "1".
- "Link yang dilalui": A dropdown menu showing "1001-1101".

 Each field's navigation bar includes arrows for navigation and a circular refresh icon. A "Selesai" button is located at the bottom right of the window.

Gambar 4.37 Tampilan Masukan Data Trayek

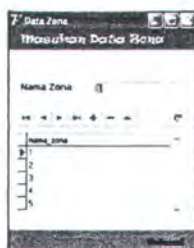
4.3.2.3. Tampilan Masukan Survey HI

Tampilan masukan survey HI berfungsi untuk memasukan data-data survey HI yang dilakukan oleh Dinas Perhubungan Kota Mojokerto. Tampilan ini merupakan antarmuka dari masukan data survey HI. Tampilan ini terdiri dari 3 tampilan di dalamnya, yaitu:

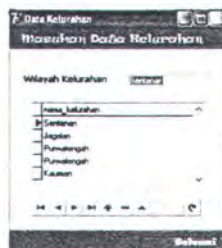
- Tampilan Masukan Survey HI, merupakan antarmuka untuk masukan Survey HI. Tampilan ini dapat dilihat pada gambar 4.38.
- Tampilan Masukan Asal Tujuan Perjalanan, merupakan antarmuka untuk masukan Asal Tujuan perjalanan. Tampilan ini dapat dilihat pada gambar 4.39.
- Tampilan Masukan sampel survey, merupakan antarmuka untuk masukan sampel survey HI. Tampilan ini dapat dilihat pada gambar 4.40.

This is a partial screenshot of a window titled "Form Survey Wawancara Rumah Tangga". The sub-header is "Masukan Data Survey Wawancara Rumah Tangga". The visible part of the form includes a label "Tahun" followed by a dropdown menu.

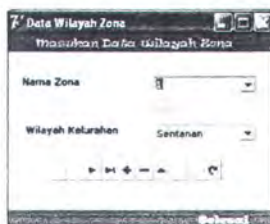
- Tampilan masukan data node (gambar 4.44), merupakan antarmuka dari masukan data node.
- Tampilan masukan data link (gambar 4.45), merupakan antarmuka dari masukan data link.
- Tampilan masukan data jalan (gambar 4.46), merupakan antarmuka dari masukan data jalan.



Gambar 4.41 Tampilan Masukan Data Zona



Gambar 4.42 Tampilan Masukan Data Kelurahan



Gambar 4.43 Tampilan Masukan Data Wilayah Zona

Gambar 4.45 Tampilan Masukan Data Link

Gambar 4.46 Tampilan Masukan Data Jalan

4.3.3. Tampilan Proses Perencanaan Ulang Trayek Angkutan

Proses Perencanaan Ulang Trayek Angkutan mempunyai dua jenis tampilan, yaitu tampilan untuk masukan data tahun perencanaan dan tampilan untuk hasil proses perencanaan.

4.3.3.1. Tampilan Masukan Tahun Untuk Proses Perencanaan Ulang

Proses Perencanaan Trayek Angkutan Kota

Perancangan Untuk Tahun

Menggunakan Data Tahun

Gambar 4.47 Tampilan Untuk Masukan Tahun Perencanaan

4. 3. 3. 2. Hasil Laporan Proses Perencanaan Ulang Trayek Angkutan

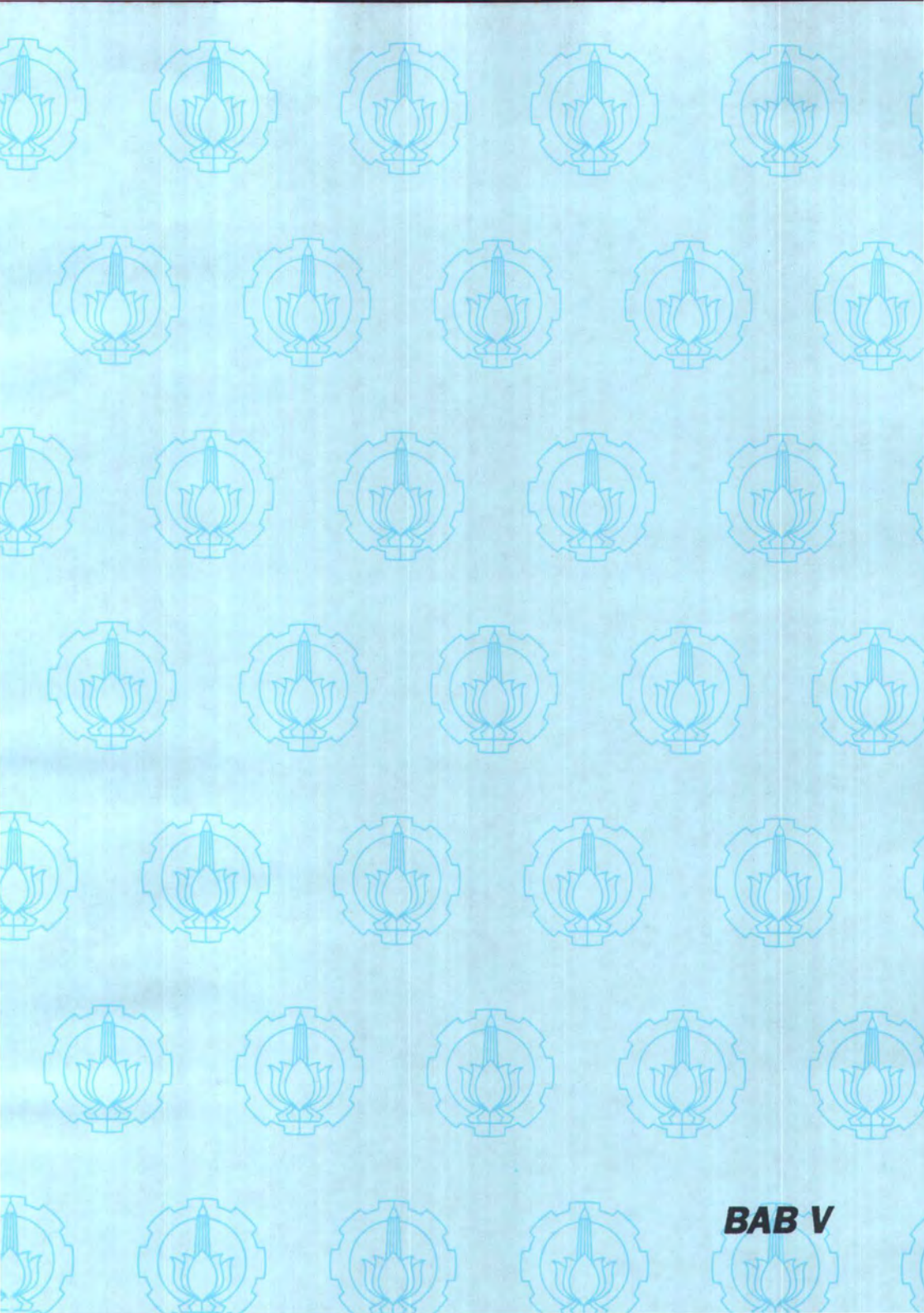
Tampilan hasil laporan proses perencanaan ulang trayek angkutan merupakan tampilan untuk menampilkan keluaran proses perencanaan trayek angkutan kota. Tampilan ini terdiri dari banyak tampilan yang berbeda-beda sesuai dengan jenis laporannya, tetapi pada dasarnya semua laporan yang dihasilkan mempunyai bentuk tampilan yang sama, seperti dapat dilihat pada gambar 4.48.

Faktor Pertumbuhan Data Kependudukan

Faktor Pertumbuhan Untuk Data Jumlah Penduduk Adalah 0.00936233100267934

Faktor Pertumbuhan Untuk Data Jumlah Pendapatan Adalah 0.143313894304285

Faktor Pertumbuhan Untuk Data Kepemilikan Kendaraan Adalah 0.146489210471791



BAB V

BAB V

UJI COBA DAN EVALUASI

Pada bab ini dibahas tentang pelaksanaan uji coba dan evaluasi yang dilakukan. Uji coba yang dilakukan didasarkan pada tujuan pembuatan sistem. Uji coba dilakukan pada setiap tahapan-tahapan proses aplikasi sampai tercapai tujuan yang diinginkan.

5.1. LINGKUNGAN UJI COBA APLIKASI

Lingkungan pengujian aplikasi meliputi lingkungan perangkat keras dan perangkat lunak yang digunakan. Perangkat keras yang digunakan adalah processor Pentium IV 2.80 GHz dengan memori 256 MB RAM. Sedangkan perangkat lunaknya digunakan sistem operasi *Microsoft Windows XP*, perangkat lunak pembangun aplikasi digunakan *Borland Delphi 7*, perangkat lunak sebagai tempat penyimpanan basis-data digunakan *Microsoft SQL Server 2000*. Selain itu digunakan perangkat lunak tambahan untuk menguji kebenaran terhadap hasil proses yang dilakukan, digunakan SPSS 13.0 (*Statistical Product and Service Solution 13.0*).

5.2. DATA UJI COBA APLIKASI

Data yang digunakan dalam uji coba adalah data sekunder yang berasal dari Dinas Perhubungan Kota Mojokerto. Data-data tersebut merupakan data hasil survey yang dilakukan oleh pihak Dinas Perhubungan Kota Mojokerto yang telah diakumulasi, yaitu data hasil survey HI, data jumlah sampel yang diambil, data jaringan jalan kota mojokerto, data wilayah kota mojokerto, data trayek angkutan kota, serta data kependudukan untuk jangka waktu beberapa tahun. Contoh dari

- a. Uji coba verifikasi terhadap tiap tahapan proses dari aplikasi yang dibuat, untuk membuktikan bahwa hasil proses aplikasi yang dibuat tersebut adalah benar.
- b. Uji coba validasi terhadap hasil akhir dari proses aplikasi yang dibuat, yang membuktikan bahwa hasil yang diperoleh telah sesuai dengan analisis kebutuhan yang diinginkan.

5. 4. PELAKSANAAN UJI COBA DAN ANALISIS HASIL VERIFIKASI TIAP TAHAPAN PROSES APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA (SKENARIO 1 SAMPAI SKENARIO 6)

Terdapat beberapa tahapan proses yang harus diuji hasil kebenaran prosesnya, karena tahapan proses tersebut yang menentukan apakah hasil perhitungan akhir dari aplikasi telah benar dan sesuai dengan kenyataan. Uji tiap tahapan proses tersebut selanjutnya akan dibahas pada sub-bab berikut.

5. 4. 1. Pelaksanaan Uji Coba Dan Analisis Hasil Proses Mencari Persamaan Regresi Untuk Menghitung Bangkitan Perjalanan Tiap Zona (Skenario 1)

Persamaan Regresi digunakan untuk melihat keterkaitan antara jumlah perjalanan, yang merupakan variabel terikat, dengan beberapa faktor kependudukan seperti jumlah penduduk, jumlah pendapatan, dan jumlah kepemilikan kendaraan, yang merupakan variabel bebasnya. Dan berikut akan dijelaskan uji coba yang dilakukan.

5. 4. 1. 1. Langkah Pelaksanaan Uji Coba Skenario 1

4.1.2. Hasil Uji Coba Skenario 1

Hasil uji coba yang dilakukan menghasilkan persamaan regresi linier. Hal ini dapat dilihat pada gambar 5.1. Dari hasil keluaran aplikasi perencanaan tersebut dapat dilihat bahwa jumlah perjalanan yang merupakan variabel terikat sangat bergantung pada jumlah penduduk yang merupakan variabel bebasnya.

Persamaan Regresi Untuk Melihat Keterkaitan Antara Jumlah Perjalanan Dengan Data Kependudukan

variabel Y = jumlah perjalanan

variabel x1 = jumlah penduduk

variabel x2 = jumlah pendapatan

variabel x3 = jumlah kendaraan pribadi

Persamaan Y = (7505.4100905847) + (0.808899917342939x1) + (0x2) + (0x3)

Nilai F = 184.927765300374

Nilai R-Kuadrat = 0.924595744704866

Nilai t untuk konstanta = 15.5500690721125

Nilai t untuk koefisien x1 = 13.59881484911

Nilai t untuk koefisien x2 = 0

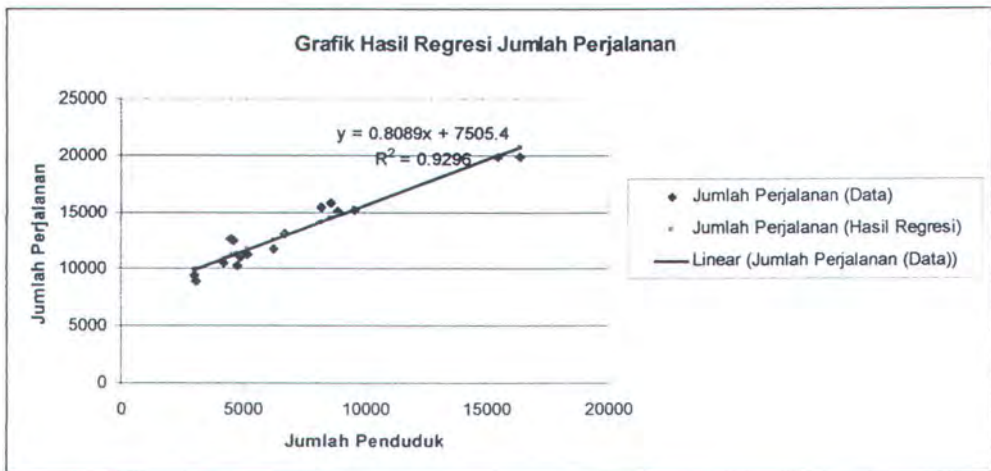
Nilai t untuk koefisien x3 = 0

Gambar 5.1 Hasil Proses Mencari Persamaan Regresi Jumlah Perjalanan

Persentase kesalahan antara data asli dan data hasil proses regresi dapat dilihat pada tabel 5.1, dan grafiknya yang dapat dilihat pada gambar 5.2. Dengan demikian kesalahan antara 0% sampai 11%.

Tabel 5.1 Persentase Kesalahan Antara Data Asli Dan Data Hasil Regresi

Persentase Perbandingan Data Asli Dengan Data Hasil Regresi				
Zona	Data Jumlah Penduduk (x)	Data Jumlah Perjalanan (y)	Jumlah Perjalanan Hasil Regresi (y')	Persentase Kesalahan (dalam %)



Gambar 5.2 Grafik Perbandingan Antara Data Asli Dan Data Hasil Regresi Jumlah Perjalanan

5. 4. 1. 3. Analisis Hasil Skenario 1

Berdasarkan pada hasil uji coba yang dilakukan, terbukti bahwa hasil proses mencari persamaan regresi telah benar. Dengan hasil perhitungan yang sama antara luaran aplikasi untuk perencanaan trayek dengan hasil luaran SPSS 13.0, yang dapat dilihat pada lampiran E.

Berdasarkan pada persen kesalahan yang ada, membuktikan bahwa persen kesalahan tidak terlalu tinggi yaitu antara 1% sampai dengan 11%, sehingga persamaan regresi linier ini dapat digunakan untuk mendapatkan model mencari bangkitan perjalanan.

5. 4. 2. Pelaksanaan Uji Coba Dan Analisis Hasil Faktor Ekspansi (Skenario 2)

Faktor Ekspansi Data Sampel Tiap Zona
Faktor Ekspansi Zona 1 adalah 20
Faktor Ekspansi Zona 2 adalah 20
Faktor Ekspansi Zona 3 adalah 20
Faktor Ekspansi Zona 4 adalah 20
Faktor Ekspansi Zona 5 adalah 20
Faktor Ekspansi Zona 6 adalah 20
Faktor Ekspansi Zona 7 adalah 20
Faktor Ekspansi Zona 8 adalah 20
Faktor Ekspansi Zona 9 adalah 20
Faktor Ekspansi Zona 10 adalah 20
Faktor Ekspansi Zona 11 adalah 20
Faktor Ekspansi Zona 12 adalah 20
Faktor Ekspansi Zona 13 adalah 20
Faktor Ekspansi Zona 14 adalah 20
Faktor Ekspansi Zona 15 adalah 20
Faktor Ekspansi Zona 16 adalah 20

Gambar 5.3 Faktor Ekspansi Tiap Zona

Bangkitan Perjalanan Untuk Zona 1 adalah 11800
Bangkitan Perjalanan Untuk Zona 2 adalah 13100
Bangkitan Perjalanan Untuk Zona 3 adalah 15020
Bangkitan Perjalanan Untuk Zona 4 adalah 15400
Bangkitan Perjalanan Untuk Zona 5 adalah 19900
Bangkitan Perjalanan Untuk Zona 6 adalah 15200
Bangkitan Perjalanan Untuk Zona 7 adalah 19800
Bangkitan Perjalanan Untuk Zona 8 adalah 15800
Bangkitan Perjalanan Untuk Zona 9 adalah 10520
Bangkitan Perjalanan Untuk Zona 10 adalah 12600
Bangkitan Perjalanan Untuk Zona 11 adalah 10300
Bangkitan Perjalanan Untuk Zona 12 adalah 12520
Bangkitan Perjalanan Untuk Zona 13 adalah 9000
Bangkitan Perjalanan Untuk Zona 14 adalah 9400
Bangkitan Perjalanan Untuk Zona 15 adalah 11220
Bangkitan Perjalanan Untuk Zona 16 adalah 10980

Gambar 5.4 Bangkitan Perjalanan Dalam Populasi

5. 4. 2. 3. Analisis Hasil Skenario 2

Dari hasil uji coba yang dilakukan membuktikan bahwa proses mencari faktor ekspansi dan data populasinya telah benar, hal ini dapat dilihat hasil yang diperoleh dengan salah satu perhitungan manual yang dilakukan mempunyai hasil

Tabel 5.2 Perbandingan Data Asli Dengan Data Peramalan Untuk Menghitung Data Jumlah Penduduk

Faktor Pertumbuhan Data Jumlah Penduduk					
Tahun	Jumlah Penduduk	Faktor Pertumbuhan	Faktor Pertumbuhan Rata-Rata	Jumlah Penduduk Hasil Peramalan	Persen Kesalahan
1998	107123	-	-	-	-
1999	108027	0.008438897	0.008438897	-	-
2000	108958	0.007692521	0.008065709	108.939	(0.07407)
2001	109911	0.009673152	0.008601524	109.736	0.15920
2002	111249	0.012173486	0.009494514	110.856	0.36290
2003	112547	0.011667521	0.009929115	112.305	0.21479
2004	113275	0.006468409	0.009352331	113.864	(0.34385)

Tabel 5.3 Perbandingan Data Asli Dengan Data Peramalan Untuk Menghitung Data Jumlah Pendapatan

Faktor Pertumbuhan Data Jumlah Pendapatan					
Tahun	Jumlah Pendapatan	Faktor Pertumbuhan	Faktor Pertumbuhan Rata-Rata	Jumlah Pendapatan Hasil Peramalan	Persen Kesalahan
2000	1862363152	-	-	-	-
2001	2182665381	0.159529375	0.159529375	-	-
2002	2489200083	0.140445764	0.149987569	2.530.853.030	(1.67395)
2003	2626500356	0.135505488	0.145160209	2.862.549.153	(1.27539)
2004	3215921301	0.13777495	0.143313894	3.236.795.739	(0.64910)

Tabel 5.4 Perbandingan Data Asli Dengan Data Peramalan Untuk Menghitung Data Jumlah Kendaraan

Faktor Pertumbuhan Data Jumlah Kendaraan					
Tahun	Jumlah Kendaraan	Faktor Pertumbuhan	Faktor Pertumbuhan Rata-Rata	Jumlah Kendaraan Hasil Peramalan	Persen Kesalahan
2002	23312	-	-	-	-
2003	26562	0.139413178	0.139413178	-	-
2004	30641	0.153665244	0.146489211	30.265	1.22691

Kemudian dari hasil faktor pertumbuhan tiap tahunnya, maka dapat diperoleh faktor pertumbuhan rata-rata, yang nantinya digunakan untuk peramalan data kependudukan pada tahun rencana, seperti dapat dilihat pada gambar 5.5, yang merupakan faktor pertumbuhan untuk masing-masing data kependudukan hasil dari proses aplikasi.

Faktor Pertumbuhan Untuk Data Jumlah Penduduk Adalah 0.00935233100287934
 Faktor Pertumbuhan Untuk Data Jumlah Pendapatan Adalah 0.143313894304285
 Faktor Pertumbuhan Untuk Data Kepemilikan Kendaraan Adalah 0.146489210671791

encana, seperti dapat dilihat pada tabel 5.5, yang akan dibandingkan dengan ramalan data kependudukan pada suatu tahun rencana hasil proses aplikasi tertentu pada tabel 5.6.

Tabel 5.5 Perhitungan Manual Untuk Mendapatkan Data Kependudukan Dan Perjalanan Pada Tahun Rencana

Parameter	Nilai
Jumlah Penduduk	6240
Faktor Pertumbuhan Jumlah Penduduk	0.009
Jumlah Penduduk Tahun Rencana	$6240 * (1 + 0.009)^{2006-2005} = 6298.358$
Jumlah Pendapatan	181304980
Faktor Pertumbuhan Jumlah Pendapatan	0.143
Jumlah Pendapatan Tahun Rencana	$181304980 * (1 + 0.143)^{2006-2005} = 207288502.685$
Jumlah Kendaraan	1780
Faktor Pertumbuhan Jumlah Kendaraan	0.146
Jumlah Kendaraan Tahun Rencana	$1780 * (1 + 0.146)^{2006-2005} = 2040.751$
Jumlah Bangkitan Perjalanan Tahun Rencana	$7505.4 + (0.8089 * 6298.358) = 12600.142$

Tabel 5.6 Tabel Hasil Proses Aplikasi Untuk Meramalkan Data Kependudukan Dan Bangkitan Perjalanan Pada Suatu Tahun Rencana

	Jumlah Penduduk	Jumlah Pendapatan	Jumlah Kendaraan	Jumlah Bangkitan Perjalanan
Tahun 2005	6240	181304980	1780	7505.4
Tahun 2006	6298.358	207288502.685	2040.751	12600.142

Zona 8	8660.2430	393013419.446205	2774.50389	14511
Zona 9	4259.46684	175117044.597789	1398.71684	10951
Zona 10	4562.2725	183638140.185761	1467.50619	11196
Zona 11	4804.5171	171928890.937244	1696.80403	11392
Zona 12	4683.3948	189898515.477969	1605.08489	11294
Zona 13	3108.8052	130946507.134294	1100.6296	10020
Zona 14	3028.05699	129265469.84922	894.26158	9955
Zona 15	5248.6321	177261809.998392	1719.7338	11751
Zona 16	4925.639375	157379261.24855	1398.7168	11490

4.3.3. Analisis Hasil Skenario 3

Berdasarkan pada hasil uji coba yang dilakukan dengan melihat persen kesalahan yang ada, yaitu 0% sampai 0.4% untuk peramalan jumlah penduduk, 0.5% sampai 1.7% untuk jumlah pendapatan dan 1.22% untuk jumlah kendaraan, maka faktor pertumbuhan dapat digunakan untuk melakukan peramalan data penduduk jumlah penduduk, jumlah pendapatan dan jumlah kendaraan.

Selain itu dengan membandingkan hasil perhitungan manual yang dilakukan untuk meramalkan data kependudukan menggunakan rumus tingkat pertumbuhan dan meramalkan data bangkitan perjalanan menggunakan persamaan regresi, dengan peramalan hasil proses aplikasi, diperoleh hasil yang sama, sehingga hasil proses akhir untuk meramalkan data kependudukan dan bangkitan perjalanan terbukti benar.

4.4. Pelaksanaan Uji Coba Dan Analisis Hasil Metode Furnees (Skenario 4)

regresi dengan nilai bangkitan perjalanan tiap zona hasil proses aplikasi. Dimana nilai perbandingan tersebut harus mempunyai nilai satu atau mendekati satu. Atau dalam hal ini nilai dengan interval antara 0.999 sampai 1.001.

5. 4. 4. 2. Hasil Uji Coba Skenario 4

Hasil uji coba dari pencarian jumlah perjalanan antar zona, akan diperoleh beberapa iterasi perhitungan pencarian jumlah perjalanan antar zona. Dengan menggunakan data awal seperti dapat dilihat pada tabel 5.7, sampai dengan diperoleh hasil perhitungan akhir seperti dapat dilihat pada tabel 5.8. Sedangkan tahapan tiap proses iterasinya dari awal sampai akhir dapat dilihat pada lampiran B.

Tabel 5.7 Data Awal Perjalanan Antar Zona

O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	i	i	E (I/I)
1	0	1340	1160	1280	1540	620	2280	1000	180	560	260	460	100	180	480	180	11800	12600	1.067797
2	2000	0	1320	1140	1540	740	1580	900	500	600	340	820	240	220	660	480	13100	13006	0.992577
3	1780	1340	0	1580	1580	680	2420	720	360	1300	420	900	220	380	620	580	15020	14772	0.983495
4	1620	1160	1300	0	1560	1020	2240	1060	680	1460	800	740	280	280	500	420	15400	14194	0.921029
5	2300	1520	1620	1700	0	1260	2240	1260	680	1420	1420	1980	300	340	1260	520	19900	20095	1.009795
6	1500	980	920	1240	1420	0	1620	840	740	1180	860	1040	600	540	980	740	15200	15343	1.009408
7	2900	1600	2420	2300	2220	1420	0	720	500	980	1020	960	520	420	1340	480	19800	20863	1.053667
8	1980	1260	1060	1340	1580	820	980	0	900	1540	700	920	520	480	1000	720	15800	14511	0.918418
9	1120	760	580	960	940	780	740	920	0	580	560	680	460	440	600	400	10520	10951	1.04097
10	1200	620	1240	1460	1400	920	920	1200	260	0	720	900	360	360	580	440	12600	11196	0.888571
11	940	480	560	920	1480	680	1020	500	400	900	0	680	340	340	660	400	10300	11392	1.106019
12	1040	860	900	880	1980	860	900	760	460	1020	660	0	500	420	920	360	12520	11294	0.902077
13	920	540	460	600	620	640	680	500	440	620	560	680	0	600	820	320	9000	10020	1.113333
14	1060	560	560	600	660	600	660	480	480	700	600	720	600	0	720	380	9400	9955	1.059043
15	1060	660	860	600	1340	780	1300	800	420	640	640	880	560	400	0	280	11220	11751	1.047326
16	1140	800	880	880	900	840	760	740	500	800	680	660	440	400	580	0	10980	11490	1.046446
d	22760	14480	15620	17400	20880	12660	20340	12480	7500	14320	10240	13020	6120	5600	11940	6700	212560	219425	1.040669
E	12600	13008	14772	14184	20095	15343	20863	14511	10951	11196	11392	11294	10020	9955	11751	11490	213425	219425	1.000000
E (I/I)	0.954	0.898	0.928	0.815	0.962	1.212	1.026	1.163	1.46	0.782	1.113	0.667	1.637	1.72	0.984	1.715	1.00407		

Tabel 5.8 Iterasi Keempat Perhitungan Perjalanan Antar Zona Untuk Suatu Tahun Rencana

ITERASI 4																			
O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	i	i	E (I/I)
1	0	1272	1138	1097	1681	794.3	2486	1332	279.8	464.2	307.8	419.4	174.4	327	498.1	309.2	12600.7	12600	0.999942
2	1118	0	1236	932.1	1506	904.6	1644	1059	741.7	474.6	384.1	713.3	399.3	382	673.4	837.7	13006.1	13008	1.000148
3	989.3	1207	0	1219	1536	626.2	2503	842.2	530.8	1038	471.6	778.2	363.8	695	807.1	1006	14772.6	14772	0.999945
4	925.3	955.5	1175	0	1387	1134	2119	1134	917.2	1050	821.8	585.3	423.6	442	450.2	666.5	14186.1	14184	0.999955
5	1307	1400	1542	1413	0	1566	2369	1507	1025	1142	1631	1751	642.8	600	1269	922.6	20095.1	20095	1.000073
6	816.3	864.1	838.3	986.8	1352	0	1641	962.2	1068	908.4	945.7	880.5	917.1	912	944.5	1257	15348.2	15343	0.999661
7	1798	1571	2456	2038	2353	1882	0	918.5	804	840.2	1249	905.2	937.9	790	1438	908	20863.5	20863	1.000049
8	1004	1036	900.3	994	1402	909.4	925.1	0	1211	1105	717.5	726	784.9	756	868.4	1140	14509.1	14511	1.00013
9	630.9	693.6	547.1	790.8	926.1	960.6	775.7	1091	0	462.2	637.4	595.9	771.1	769	598.6	703.3	10953.3	10951	0.999786
10	577.2	483.2	998.6	1041	1178	967.5	823.5	1215	331.8	0	699.7	673.5	515.3	537	494.1	660.6	11195.8	11196	1.000014
11	566.6	468.8	565.2	810.9	1590	896.1	1144	634.3	639.7	767.4	0	637.7	609.9	636	704.6	752.5	11394.2	11392	0.99981
12	502.4	673.1	728	621.6	1673	908.3	809.1	772.7	589.6	897	644.2	0	718.8	630	787.1	542.8	11297.4	11294	0.999703
13	640.1	873.3	468.7	670.6	547.3	926.1	785.2	626.1	506.2	572.5	675.2	631.4	0	1113	866.7	506.1	10021.7	10020	0.999828

dengan menggunakan persamaan regresi dengan jumlah bangkitan perjalanan hasil proses aplikasi adalah hampir mendekati nilai satu. Dimana hasil akhir tersebut diperoleh pada hasil proses iterasi yang keempat. Hal ini membuktikan bahwa hasil perhitungan peramalan jumlah perjalanan antar zona telah benar.

5.4.5. Pelaksanaan Uji Coba Dan Analisis Hasil Algoritma *Dijkstra* (Skenario 5)

Algoritma *Dijkstra* digunakan untuk mendapatkan rute tercepat antara dua titik pusat zona. Rute ini nantinya dianggap sebagai rute yang dilalui oleh kendaraan pribadi untuk perjalanan antar zona yang dilakukan. Sebagai asumsi awal bahwa para pengguna jalan yang menggunakan kendaraan pribadi akan selalu memilih jalur tecepat dari suatu jaringan jalan untuk berpindah dari satu pusat zona ke pusat zona lainnya.

5.4.5.1. Langkah Pelaksanaan Uji Coba Skenario 5

Uji coba algoritma *dijkstra* dilakukan dengan menggunakan beberapa contoh jaringan jalan baru, yang telah diberi bobot pada masing-masing *link*-nya. Bila hasil yang dikeluarkan aplikasi untuk menunjukkan rute terpendek dari tiap jaringan jalan tersebut dari dua buah titik simpul benar, maka algoritma *dijkstra* pada aplikasi untuk perencanaan trayek angkutan kota juga benar. Contoh jaringan jalan tersebut dapat dilihat pada lampiran F.

5.4.5.2. Hasil Uji Coba Skenario 5

Hasil dari pengujian algoritma *dijkstra* dengan menggunakan beberapa contoh jaringan jalan baru, dapat dilihat pada lampiran F. Pada hasil tersebut ditunjukkan *node-node* yang menghubungkan antara *node* awal dan *node* akhir

jaringan jalan yang diberikan, untuk dicari alternatif rute terpendeknya diantara dua buah titik yang diberikan, dan algoritma *dijkstra* tersebut memberikan alternatif rute terpendek yang benar.

5. 4. 6. Pelaksanaan Uji Coba Dan Analisis Hasil Pencarian Persamaan Regresi Persentase Penggunaan Angkutan (Skenario 6)

Persamaan regresi persentase penggunaan angkutan digunakan untuk melihat keterkaitan antara persentase penggunaan angkutan dengan data-data variabel bebasnya, seperti data jumlah penduduk (x_1), jumlah pendapatan (x_2), jumlah kendaraan (x_3), log dari RWP (x_4), log dari data RBP (x_5), dan panjang lintasan (x_6). Dimana data-data yang digunakan untuk mendapatkan persamaan regresi dikelompokkan berdasarkan perpotongan zona-zona yang dilalui oleh tiap trayek, dan dihitung persamaan regresi untuk setiap trayek yang ada.

5. 4. 6. 1. Langkah Pelaksanaan Uji Coba Skenario 6

Pelaksanaan uji coba pencarian persamaan regresi untuk mendapatkan persentase penggunaan angkutan kota dilakukan dengan melihat apakah persamaan regresi tersebut dapat digunakan untuk mencari persentase penggunaan angkutan kota yang ada. Hal ini dilakukan dengan melihat persentase kesalahan antara nilai data dengan nilai hasil persamaan regresi. Uji coba tersebut dilakukan hanya pada perhitungan trayek A saja, sedangkan perhitungan trayek lainnya dianggap telah memenuhi bila trayek A tersebut memenuhi.

Sedangkan pelaksanaan uji coba untuk melihat apakah persamaan regresi yang dihasilkan telah benar menurut hasil perhitungan tidak dilakukan, karena hal ini telah terbukti pada uji coba sebelumnya, yaitu persamaan regresi untuk

dengan hasil persamaan regresi. Kemudian pada gambar 5.7 dapat dilihat hasil persamaan regresi untuk seluruh trayek hasil dari proses aplikasi untuk perencanaan trayek angkutan.

Tabel 5.9 Tabel Perbandingan Persen Kesalahan Antara Data Asli Dengan Hasil Persamaan Regresi

RBP	log(RBP)	Persentase Penggunaan Angkutan(data)	Persentase Penggunaan Angkutan(hasil regresi)	Persen Kesalahan(%)
3.827751	0.582944	13.88888889	14.23218519	-2.471733341
12.12121	1.083546	8.695652174	9.170615982	-5.46208379
13.22314	1.121335	8.695652174	8.78853744	-1.068180561
6.61157	0.820305	12.39669421	11.832239	4.563272061
17.31602	1.238448	7.876386957	7.604408813	2.832554061
5.509642	0.741123	12.5	12.63283723	-1.062697898
4.98132	0.697344	13.33333333	13.07548409	1.933669341
Rata-Rata		11.04804396	11.04804396	

Persamaan Regresi Untuk Trayek 1
 $Y = 20.1263043027892 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (-10.1109577265754)x_5 + (0)x_6$
 Persamaan Regresi Untuk Trayek 2
 $Y = 21.5228675932103 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (-12.3173781693295)x_5 + (0)x_6$
 Persamaan Regresi Untuk Trayek 3
 $Y = 22.624450494949 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (-12.1942223695407)x_5 + (0)x_6$
 Persamaan Regresi Untuk Trayek 4
 $Y = 24.1933289629051 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (-15.8387494567127)x_5 + (0)x_6$
 Persamaan Regresi Untuk Trayek 5
 $Y = 21.3272552925122 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (-13.1559078724784)x_5 + (0)x_6$
 Persamaan Regresi Untuk Trayek 6
 $Y = 38.1936632703919 + (0)x_1 + (0)x_2 + (0)x_3 + (0)x_4 + (0)x_5 + (-0.00819750548943545)x_6$
 Persamaan Regresi Untuk Trayek 7
 $Y = 37.4902755750366 + (0)x_1 + (0)x_2 + (0)x_3 + (-14.537684722019)x_4 + (0)x_5 + (0)x_6$

Gambar 5.6 Hasil Persamaan Regresi Seluruh Trayek Angkutan

5. 4. 6. 3. Analisis Hasil Skenario 6

Dari hasil uji coba yang dilakukan membuktikan bahwa persamaan regresi dapat digunakan untuk mencari persentase penggunaan angkutan, dengan persen kesalahan yang tidak terlalu besar antara data asli dengan hasil regresi, yaitu antara -1% sampai dengan 5%. Selain itu dari rata-rata data asli dengan rata-rata hasil regresi diperoleh hasil yang sama yaitu 11.04804, yang berarti bahwa dari

angkutan kota bergantung sepenuhnya pada nilai RBP, kecuali trayek G yang bergantung sepenuhnya pada panjang rute trayek.

5. 5. PELAKSANAAN UJI COBA DAN ANALISIS HASIL VALIDASI APLIKASI UNTUK PERENCANAAN TRAYEK ANGKUTAN KOTA (SKENARIO 7)

Aplikasi untuk perencanaan trayek angkutan kota adalah program aplikasi untuk melakukan perencanaan terhadap trayek angkutan kota di kota Mojokerto. Untuk selanjutnya langkah-langkah uji coba, hasil dan evaluasi dari uji coba skenario untuk melihat validasi hasil dari proses aplikasi dapat dilihat pada sub-bab berikut.

5. 5. 1. Langkah Pelaksanaan Uji Coba Skenario 7

Uji coba aplikasi untuk perencanaan trayek angkutan kota hanya dilakukan untuk merencanakan trayek angkutan kota pada periode satu tahun mendatang saja, yaitu tahun 2006 dengan menggunakan data tahun 2005, dengan asumsi awal tahun sekarang adalah 2005.

Dalam pelaksanaan uji coba ini akan dilihat rute baru yang dihasilkan dan mengevaluasi rute baru tersebut, dengan membandingkan rute baru yang dihasilkan dengan rute lama dari tiap-tiap trayek yang ada. Kemudian akan diputuskan rute mana yang akan digunakan dari hasil evaluasi yang didapat, apakah tetap menggunakan rute lama atau menggunakan rute baru. Evaluasi yang dilakukan tersebut adalah evaluasi RWP (Rasio Waktu Perjalanan), RBP (Rasio Biaya Perjalanan), persentase penggunaan angkutan kota, dan pangsa pasarnya.

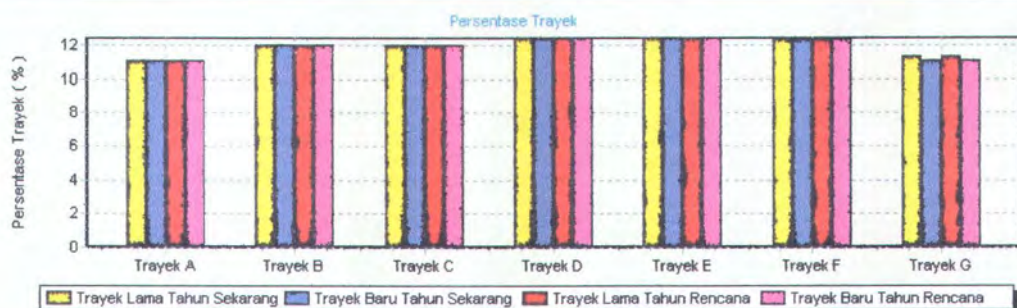
Kemudian untuk mencari nilai RWP terdapat beberapa asumsi awal dan cara perhitungan sebagai berikut:

- d. Jarak perjalanan dihitung dalam satuan meter, dan kecepatan dihitung dengan satuan meter/menit.
- e. Data kecepatan kendaraan tersedia dalam satuan km/jam, maka untuk mendapatkan kecepatan meter/menit harus dikalikan dengan $60/1000 = 0.06$.

Sedangkan untuk mencari nilai RBP diperlukan beberapa asumsi awal yang akan digunakan dalam perhitungannya, yaitu:

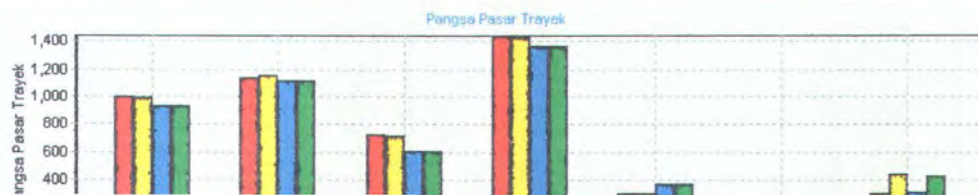
- a. Biaya untuk kendaraan pribadi :
 - Bensin yang digunakan untuk 1 km perjalanan adalah 0.1 liter (1 liter = 10 km), sehingga biaya pembelian bensin untuk 1 km perjalanan adalah Rp 450, dengan harga 1 liter = Rp 4.500.
 - Oli yang digunakan untuk 1 km perjalanan adalah 0.002 liter (1 liter = 500 km), sehingga biaya pembelian oli untuk 1 km perjalanan adalah Rp 100, dengan harga 1 liter oli = Rp 50.000.
 - Maka asumsi total biaya yang dikenakan untuk tiap 1 km perjalanan adalah $Rp\ 100 + Rp\ 450 = Rp\ 550$.
 - Sedangkan perjalanan untuk tiap 1 meternya adalah $Rp\ 550 / 1000\ m = Rp\ 0.55$.
- b. Biaya untuk angkutan umum, tarif yang dikenakan untuk semua jenis trayek angkutan umum, dengan jarak jauh maupun jarak dekat dianggap sama. Diasumsikan pula tidak terdapat pengelompokkan terhadap pengguna angkutan umum. Dengan tarif angkutan umum sebesar Rp 2000.
- c. Biaya pada tahun rencana dianggap sama dengan biaya pada tahun sekarang.
- d. RBP diperoleh dengan membandingkan biaya total menggunakan angkutan dengan biaya total menggunakan kendaraan pribadi.

	Rute Lama	Rute Baru	Rute Lama Tahun Mendatang	Rute Baru Tahun Mendatang
Trayek A	11.0480439630637	11.048043963063	11.0480439630637	11.0480439630637
Trayek B	11.9930430734579	11.993043073457	11.9930430734579	11.9930430734579
Trayek C	11.9886412575366	11.988641257536	11.9886412575366	11.9886412575366
Trayek D	12.3786058183307	12.378605818330	12.3786058183307	12.3786058183307
Trayek E	12.4074074074074	12.407407407407	12.4074074074074	12.4074074074074
Trayek F	12.3305334512231	12.330533451223	12.3305334512231	12.3305334512231
Trayek G	11.3180215322203	11.087441294216	11.3180215322203	11.0874412942167



Gambar 5.9 Evaluasi Persentase Penggunaan Angkutan Kota Masing-Masing Trayek

	Rute Lama	Rute Baru	Rute Lama Tahun Mendatang	Rute Baru Tahun Mendatang
Trayek A	1000	988.106858200914	932.625691786882	932.625691786882
Trayek B	1140	1157.94679054568	1118.66556407588	1118.66556407588
Trayek C	720	717.424665493878	604.647243204689	604.647243204689
Trayek D	1440	1433.96381386853	1364.62063885193	1364.62063885193
Trayek E	300	300.157430539803	373.857290991904	373.857290991904
Trayek F	200	211.65487768936	208.795903330386	208.795903330386
Trayek G	300	444.736322968077	313.875590914684	426.925622729021



Saran Penggunaan Rute Untuk Tahun Sekarang

Untuk trayek A pilih rute baru
 Untuk trayek B pilih rute baru
 Untuk trayek C pilih rute lama
 Untuk trayek D pilih rute lama
 Untuk trayek E pilih rute baru
 Untuk trayek F pilih rute baru
 Untuk trayek G pilih rute baru

Saran Penggunaan Rute Untuk Tahun Rencana

Untuk trayek A pilih rute baru
 Untuk trayek B pilih rute baru
 Untuk trayek C pilih rute baru
 Untuk trayek D pilih rute baru
 Untuk trayek E pilih rute baru
 Untuk trayek F pilih rute baru
 Untuk trayek G pilih rute baru

Gambar 5.11 Hasil Akhir Penentuan Penggunaan Rute Trayek Angkutan Kota

5.5.3. Analisis Hasil Skenario 7

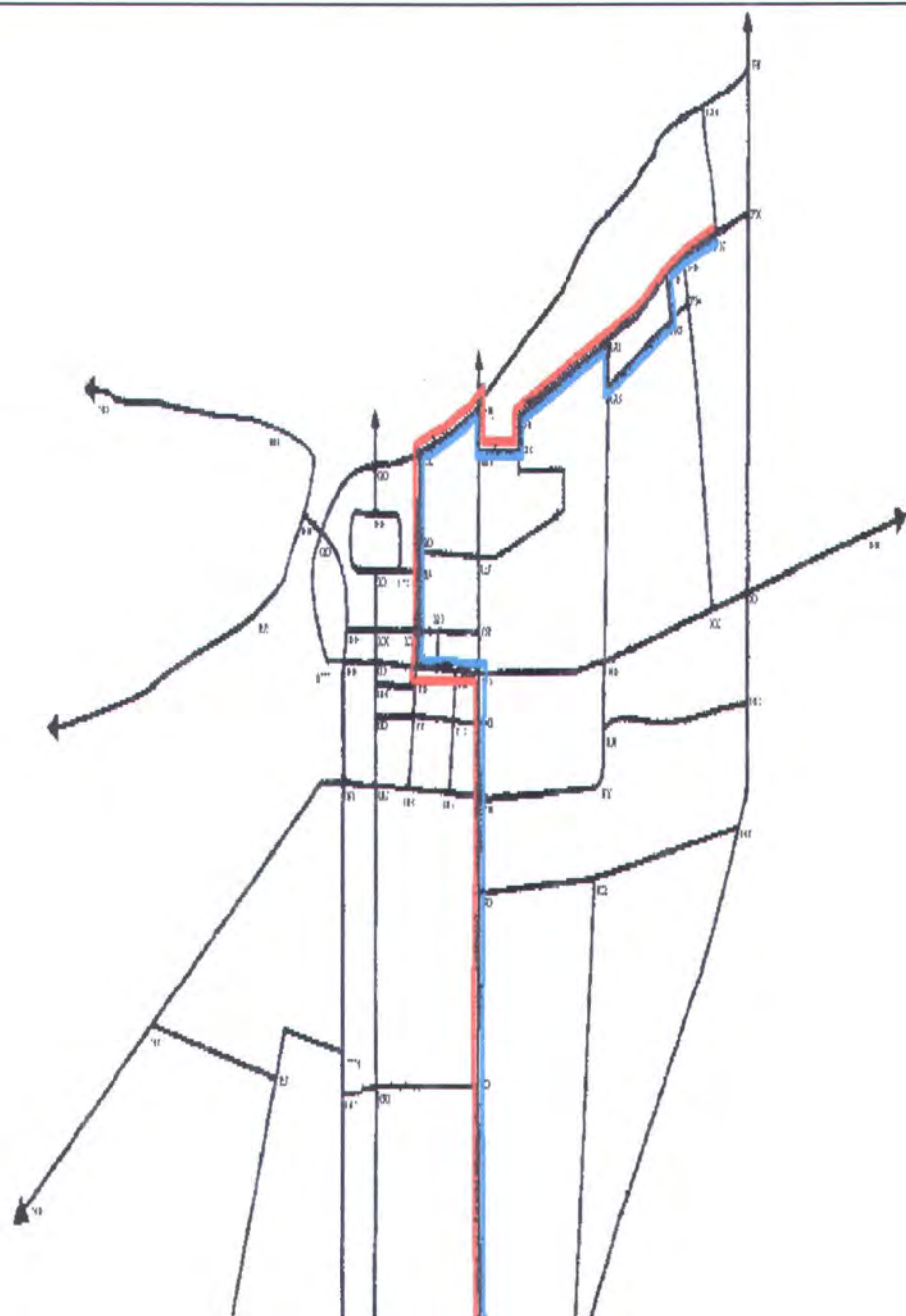
Hasil dari uji coba perencanaan rute baru trayek angkutan kota hanya trayek A dan G saja yang mengalami perubahan. Untuk lebih jelasnya dapat dilihat pada peta yang merupakan hasil penggambaran secara manual untuk memudahkan dalam melihat hasil dari proses aplikasi yang didapat, seperti ditunjukkan pada gambar 5.12 sampai 5.14 untuk arah berangkat dan pulang dari trayek A dan G, dengan warna merah merupakan rute baru dan warna biru merupakan rute lama, sedangkan rute trayek lainnya tidak mengalami perubahan (peta gambar rutenya dapat dilihat pada lampiran D), dalam artian antara rute baru dan rute lamanya mempunyai lintasan yang sama, dari hal ini dapat disimpulkan bahwa rute yang sudah ada selama ini memang sudah baik.

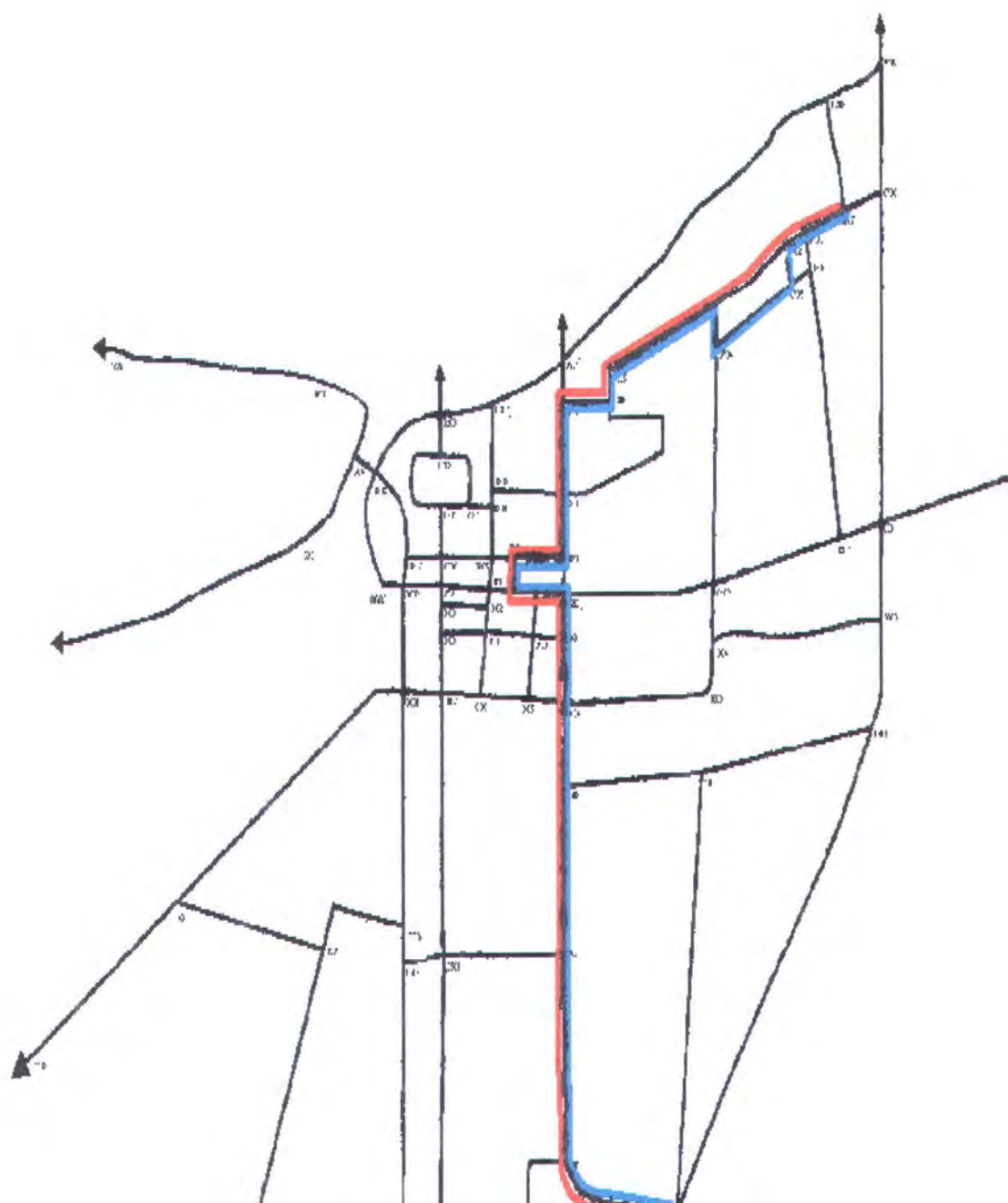
Perubahan rute baru mempunyai hasil waktu tempuh yang lebih cepat bila dibandingkan dengan rute lamanya, walaupun ada kemungkinan rute baru mempunyai lintasan yang lebih panjang dari rute, sebagai contoh rute trayek G

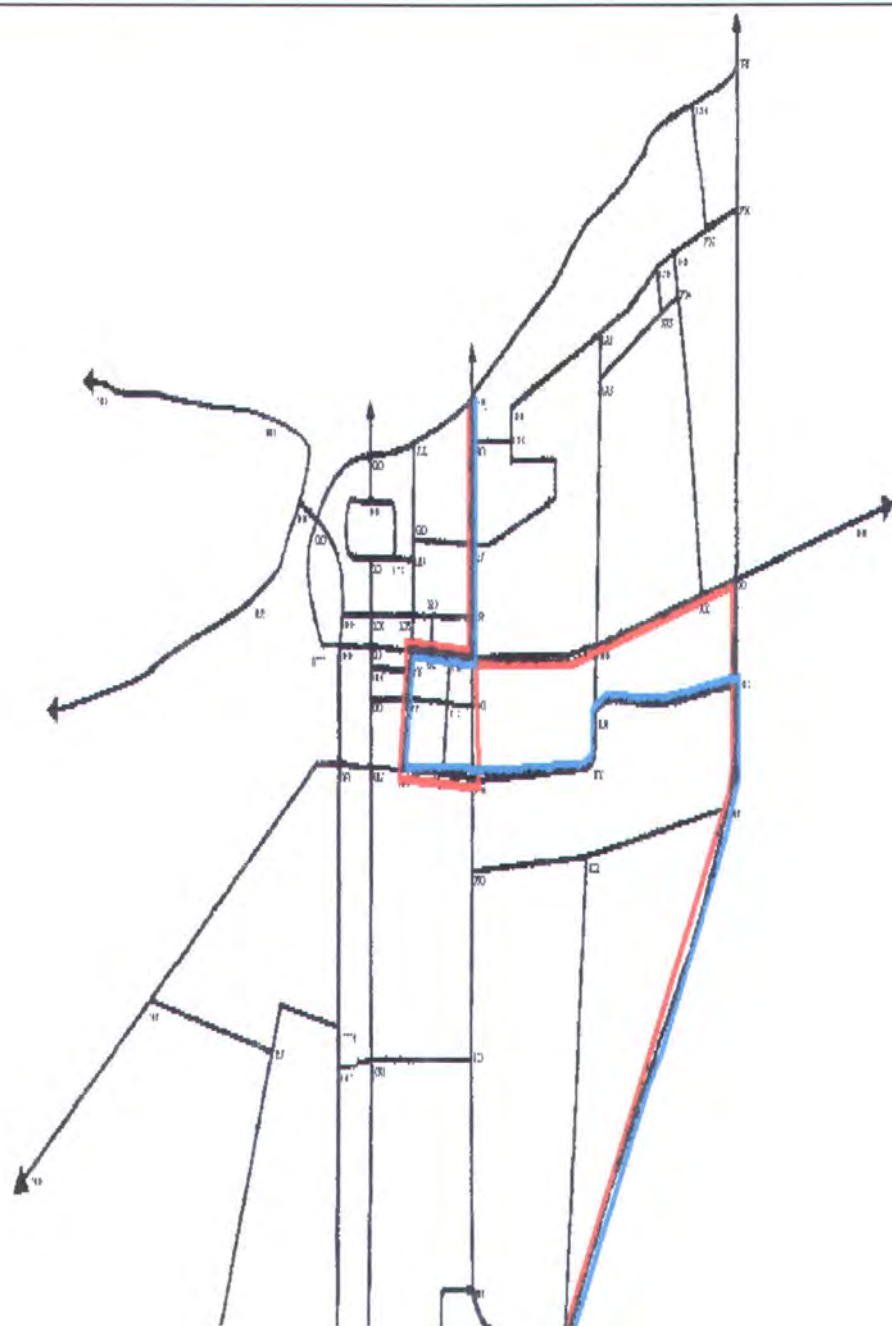
Hasil perhitungan rute lama dan rute baru selengkapnya dapat dilihat pada lampiran C.

Kemudian dapat dilihat pada gambar 5.8 di atas yang merupakan grafik perbandingan nilai RWP untuk rute lama dan rute baru hasil keluaran program, bahwa hampir semua trayek angkutan kota tidak mengalami perubahan nilai RWP, kecuali pada trayek A dan G yang mengalami penurunan nilai RWP yang berarti bahwa trayek A dan G mempunyai rasio waktu yang lebih baik bila dibandingkan dengan rute lamanya, karena semakin kecil nilai RWP maka semakin bagus rasio waktu trayek tersebut. Pada gambar 5.8 tersebut juga dapat dilihat grafik perbandingan nilai RBP antara rute lama dan rute baru, terlihat bahwa hampir seluruh trayek tidak mengalami perubahan nilai RBP, kecuali pada trayek G yang mengalami kenaikan nilai RBP, yang berarti trayek G mempunyai rasio biaya yang lebih buruk dibandingkan rute lamanya.

Setelah dilihat evaluasi terhadap nilai RWP dan RBP, dilanjutkan dengan tahap selanjutnya yaitu mengevaluasi prosentase penggunaan angkutan kota, dengan menggunakan rute lama dan rute baru untuk kondisi saat ini dan kondisi mendatang. Dimana persentase angkutan kota ini didapat dari persamaan regresi untuk mendapatkan persentase penggunaan angkutan kota. Hasil persentase penggunaan angkutan kota tersebut dapat dilihat pada gambar 5.9. dari hasil tersebut tidak terjadi perubahan persentase penggunaan angkutan kota untuk rute lama dan rute baru pada tahun 2005 dan tahun 2006, kecuali pada rute G yang mengalami penurunan persentase untuk rute barunya, hal ini dikarenakan seperti yang telah dibahas sebelumnya, model untuk mencari persentase penggunaan angkutan kota bergantung sepenuhnya pada nilai RBP dan untuk trayek G bergantung pada panjang rute, sedangkan seperti telah dilihat sebelumnya nilai







Pada gambar 5.10 merupakan pangsa pasar dari masing-masing trayek yang diperoleh dari perkalian antara persentase penggunaan masing-masing trayek dengan jumlah total perjalanan masing-masing trayek. Dan dari hasil tersebut dapat dilihat bahwa ada sebagian pangsa pasar yang naik, yaitu trayek A, C, E, F, dan G, sedangkan sisa trayek lainnya mengalami penurunan pangsa pasar untuk trayek barunya dibandingkan trayek lamanya pada tahun sekarang (tahun 2005). Sedangkan hasil pada tahun rencana pangsa pasar antara rute lama dan rute baru tidak mengalami perubahan kecuali pada trayek G yang mengalami kenaikan pangsa pasar. Dan dari keempat hasil evaluasi tersebut, dapat disimpulkan bahwa untuk trayek A dan G disarankan menggunakan rute barunya pada tahun sekarang 2005 dan tahun rencana 2006, sedangkan untuk sisa trayek lainnya, karena tidak mengalami perubahan rute, maka penggunaan rute lama maupun rute baru tidaklan menjadi suatu permasalahan. Hasil akhir dari kesimpulan penggunaan rute trayek, yang merupakan keluaran program dapat dilihat pada gambar 5.11 di atas.

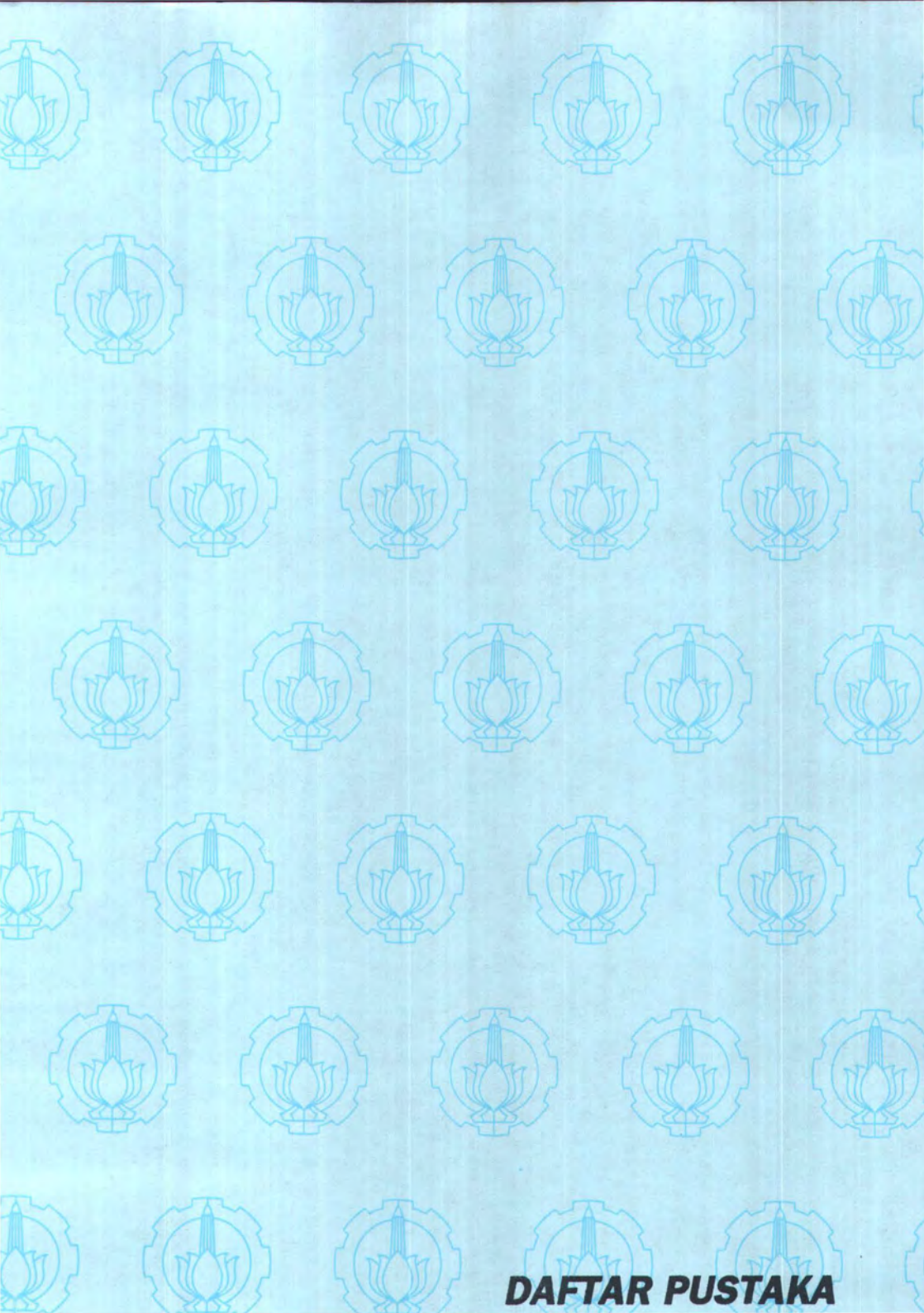
BAB VI PENUTUP

- Dan yang terakhir tahapan pemilihan moda, pada tahapan ini dilakukan evaluasi terhadap rute lama dan rute baru yang dihasilkan. Evaluasi tersebut meliputi evaluasi RWP (Rasio Waktu Perjalanan), RBP (Rasio Biaya Perjalanan), persentase penggunaan angkutan, dan pangsa pasar. Untuk memperoleh persentase penggunaan angkutan digunakan model ujung perjalanan, yang menggunakan analisis regresi linier untuk penyelesaiannya dengan variabel-variabel log dari RWP dan RBP, panjang lintasan, jumlah penduduk, jumlah pendapatan, dan jumlah kendaraan.
- b. Dari hasil uji coba verifikasi dan validasi yang dilakukan dapat diambil simpulan bahwa tiap tahapan proses dari aplikasi yang dibuat untuk mencapai tujuan akhir telah benar dan telah sesuai dengan analisis kebutuhan yang diinginkan, yaitu menghasilkan rute baru dari masing-masing trayek angkutan kota dan evaluasi terhadap masing-masing rute tersebut. Beberapa simpulan yang dapat ditarik dari hasil uji coba ini antara lain adalah sebagai berikut:
- Rute baru trayek yang dihasilkan mempunyai hasil rute yang hampir sama dengan rute yang ada selama ini. Hanya dua trayek saja yang mempunyai rute baru yang berbeda dengan rute lamanya.
 - Hasil evaluasi RWP dan RBP menunjukkan nilai yang sama hampir untuk seluruh rute lama dan rute baru trayek yang ada, hanya dua trayek saja yang mempunyai nilai RWP dan RBP yang berbeda, yaitu trayek yang mengalami perubahan rute baru terhadap rute lamanya. Nilai RWP yang diberikan mempunyai hasil yang lebih baik untuk rute barunya, tetapi nilai RBP mempunyai hasil yang lebih buruk untuk rute barunya.
 - Hasil evaluasi persentase penggunaan trayek juga memberikan nilai yang

- Hasil evaluasi pangsa pasar penggunaan angkutan kota pada tahun rencana untuk beberapa trayek angkutan kota mengalami kenaikan nilai dan beberapa trayek lainnya mengalami penurunan nilai.

6. 2. SARAN

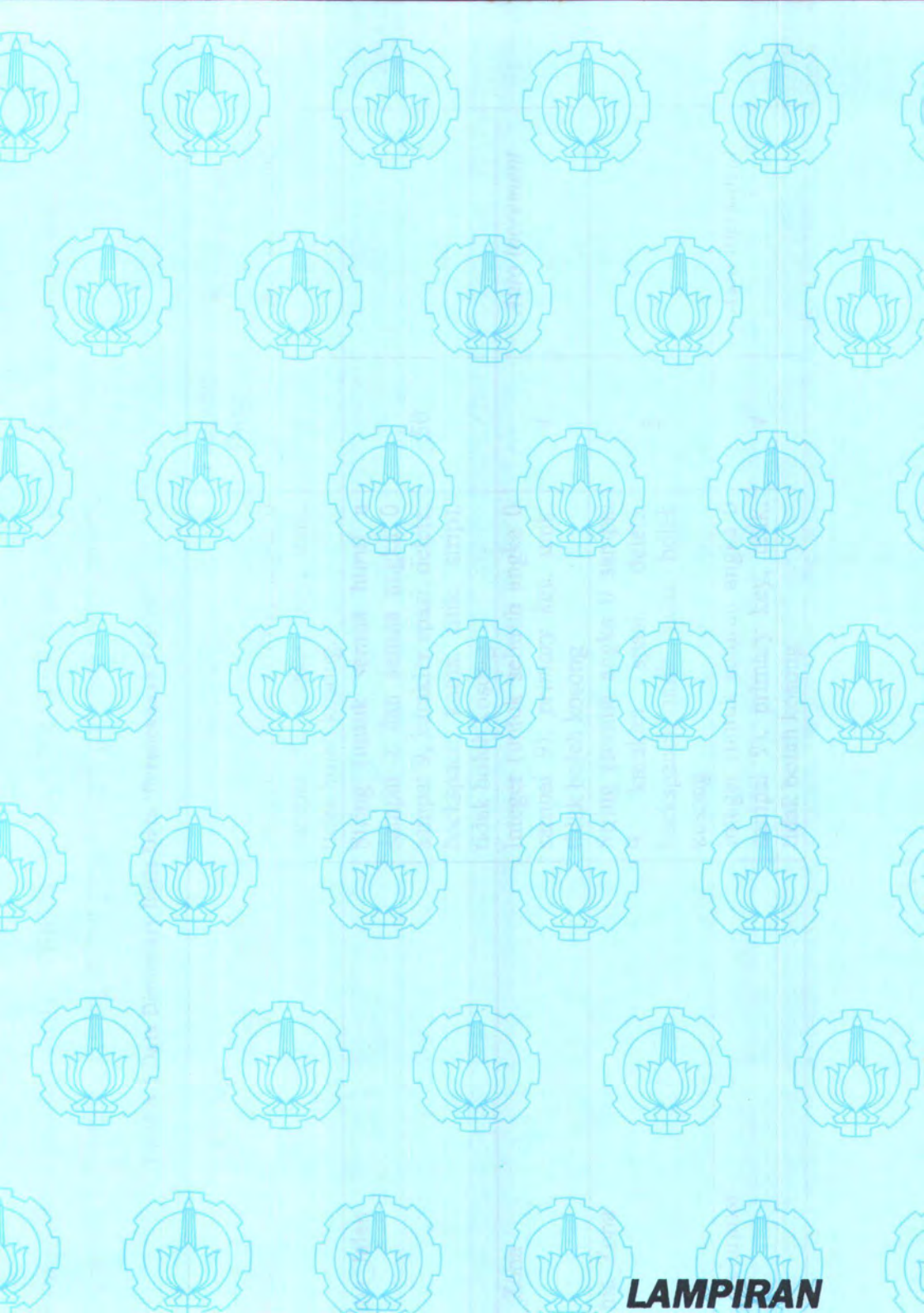
Sebuah perencanaan trayek angkutan kota yang lengkap adalah bahwa dengan hasil dari perencanaan trayek tersebut dapat diketahui jalur mana saja yang berubah, daerah mana yang harus dibukakan jalur trayek baru, serta jalur trayek mana yang harus ditutup berdasarkan faktor ekonomis (sebuah jalur angkutan kota harus benar-benar dapat memberikan pelayanan yang baik terhadap masyarakat). Sedangkan dalam aplikasi yang dibuat ini hanya dapat memberikan alternatif rute baru berdasarkan pada rute tercepat antar zona dan rute lamanya saja, yang selanjutnya dievaluasi untuk menentukan rute mana yang akan dipilih selanjutnya. Untuk itu diperlukan penambahan proses untuk dapat melengkapi proses perencanaan yang ada sehingga akan terbentuk sebuah perencanaan trayek angkutan kota yang baik, dimana dalam prosesnya dapat mempertimbangkan lahan-lahan mana yang dapat dibuka rute baru dan rute mana yang harus ditutup. Dalam hal ini diperlukan survei lapangan lebih lanjut (survey permintaan pengguna terhadap kebutuhan angkutan kota), sehingga benar-benar dapat dilihat daerah mana yang harus dibuka rute baru. Dalam survey tersebut dicari asal tujuan perjalanan penumpang angkutan kota pada titik tertentu, dan dihitung jumlah penumpang yang naik dan turun dari angkutan kota pada setiap titik-titik tersebut. Sedangkan untuk penutupan rute diperlukan data headway atau waktu antara dua buah kendaraan angkutan kota yang beroperasi secara berurutan, data *load factor* atau persentase penumpang yang telah diisi, dan data lain yang berkaitan dengan



DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [MIR-2005] Miro, Fidel, *PERENCANAAN TRANSPORTASI untuk Mahasiswa, Perencana, dan Praktisi*. Penerbit Erlangga, 2005.
- [TAM-2000] Tamin, O.Z., *Perencanaan Dan Pemodelan Transportasi*. Penerbit ITB Bandung, 2000.
- [WAL-1998] Walpole, Ronald E & Myers, Raymond H, *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuwan*. Edisi ke-4, Penerbit ITB Bandung, 1998.
- [ELM-1998] EL Masri, *Fundamental Of Basis-Data System*. Third Edition.
- [UND-1992] Undang-Undang Nomor 14 Tahun 1992 Tentang Lalu Lintas dan Angkutan Jalan.
- [PER-1993] Peraturan Pemerintah No 41 Tahun 1993 Tentang Angkutan Jalan
- [PER-1993] Peraturan Pemerintah No 43 Tahun 1993 Tentang Prasarana Dan Lalu Lintas Jalan
- [KEP-2003] Keputusan Menteri Perhubungan Nomor KM. 35 Tahun 2003 Tentang Penyelenggaraan Angkutan Orang di Jalan Dengan Kendaraan Umum
- [SUR-1996] Surat Keputusan Direktur Jenderal Perhubungan Darat Nomor 274/ HK. 105/ 96
- [TIM-2005] Tim PKL Kota Mojokerto Angkatan XXIV. *Pola Umum Lalu Lintas Dan Angkutan Jalan Dan Identifikasi Permasalahan Di Wilayah Studi Kota Mojokerto*. Laporan Umum PKL. Badan



Atribut	Tipe Data	Data	Keterangan
ia_Kelurahan	String(semua angka 0 sampai 9, karakter spasi, delete, backspace), unik, tidak boleh kosong	5	
Zona	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel zona
Kelurahan	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel kelurahan
Node	Integer(untuk seluruh angka 0 sampai 9), primary key, unik, tidak boleh kosong	4	<i>Auto increment</i>
ia_Node	String(semua angka 0 sampai 9, karakter spasi, delete, backspace), unik, tidak boleh kosong	5	
e_Pusat	Boolean(0 atau 1)	1	
Zona	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel zona
Link	Integer(untuk seluruh angka 0 sampai 9), primary key, unik, tidak boleh kosong	4	<i>Auto increment</i>

Attribut	Tipe Data	Data	Keterangan
id_Link	String(semua angka 0 sampai 9, karakter spasi, delete, backspace, strip), unik, tidak boleh kosong	5	
id_ang	Integer(untuk seluruh angka 0 sampai 9), backspace, delete	4	
id_Lalu_Lintas	Integer(untuk seluruh angka 0 sampai 9), backspace, delete	4	
id_kapasitas	Integer(untuk seluruh angka 0 sampai 9), backspace, delete	4	
id_kecepatan	Integer(untuk seluruh angka 0 sampai 9), backspace, delete	4	
id_Node_Awal	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel node
id_Node_Akhir	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel node
id_jalan	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	<i>Reference</i> tabel jalan
Total_Data_Keluarga	Integer(semua angka 0 sampai 9, karakter spasi, delete, backspace, strip), primary key 1, tidak boleh kosong	4	

Atribut	Tipe Data	Data	Keterangan
Zona	Integer(untuk seluruh angka 0 sampai 9), primary key 2, tidak boleh kosong	4	Reference tabel zona
I_Jumlah_Anggota_Seluruh_Keluarga	Float(untuk seluruh angka 0 sampai 9) , backspace, delete	8	
I_Jumlah_Pendapatan	Float(untuk seluruh angka 0 sampai 9) , backspace, delete	8	
I_Kepemilikan_Kendaraan_Pribadi	Float(untuk seluruh angka 0 sampai 9) , backspace, delete	8	
lah_Seluruh_Keluarga	Integer(untuk seluruh angka 0 sampai 9) , backspace, delete	4	
lah_Keluarga_Disurvey	Integer(untuk seluruh angka 0 sampai 9) , backspace, delete	4	
arga Menolak Diwawancara	Integer(untuk seluruh angka 0 sampai 9) , backspace, delete	4	
lah Rumah Kosong	Integer(untuk seluruh angka 0 sampai 9) , backspace, delete	4	
Zona	Integer(untuk seluruh angka 0 sampai 9), foreign key, tidak boleh kosong	4	Reference tabel zona
Zona_Asal	Integer(untuk seluruh angka 0 sampai 9), primary key 1, unik, tidak boleh kosong	4	Reference tabel zona
Zona_Tujuan	Integer(untuk seluruh angka 0 sampai 9), primary key 2, unik, tidak boleh kosong	4	Reference tabel zona

Atribut	Tipe Data	Angka Data	Keterangan
mlah_Perjalanan	Integer(untuk seluruh angka 0 sampai 9), primary key 3, unik, tidak boleh kosong	4	
mlah_Perjalanan_Angkutan	Integer(untuk seluruh angka 0 sampai 9), primary key 4, unik, tidak boleh kosong	4	
D_Trayek	Integer(untuk seluruh angka 0 sampai 9), primary key, unik, tidak boleh kosong	4	Auto increment
ama_Trayek	String(untuk semua huruf abjad a sampai z dan semua angka 0 sampai 9, karakter spasi, delete, backspace, koma, titik, strip), unik, tidak boleh kosong	5	
arif_Penumpang	Integer(untuk seluruh angka 0 sampai 9)	4	
D_Link	Integer(untuk seluruh angka 0 sampai 9), primary key 1, unik, tidak boleh kosong	4	<i>Reference</i> tabel link
D_Trayek	Integer(untuk seluruh angka 0 sampai 9), primary key 2, unik, tidak boleh kosong	4	<i>Reference</i> tabel trayek
omor_Rute	Integer(untuk seluruh angka 0 sampai 9)	4	

Atribut	Tipe Data	Data	Keterangan
0_Komponen_Kependudukan	Integer(untuk seluruh angka 0 sampai 9), primary key, unik, tidak boleh kosong		Auto increment
ama_Komponen	String(untuk semua huruf abjad a sampai z dan semua angka 0 sampai 9, karakter spasi, delete, backspace, koma, titik, strip)	50	
_Komponen_Kependudukan	Integer(untuk seluruh angka 0 sampai 9), primary key 1, unik, tidak boleh kosong	4	<i>Reference</i> tabel komponen_kependu- dukan
0_Item_Komponen_Kependudukan	Integer(untuk seluruh angka 0 sampai 9), primary key 2, unik, tidak boleh kosong	4	<i>Auto increment</i>
ahun	Integer(untuk seluruh angka 0 sampai 9),	4	
ilai	Integer(untuk seluruh angka 0 sampai 9),	4	

LAMPIRAN B

Lampiran B berisi data hasil iterasi dari metode furness untuk mencari persebaran perjalanan antar zona, dengan dimulai dari data awal, iterasi satu sampai dengan iterasi terakhir yaitu iterasi 4.

Data Tabel Asal Tujuan Perjalanan

Tabel B.1 Data Awal

O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	I	T	E(I/T)
1	0	1340	1160	1280	1640	620	2280	1090	180	560	260	460	100	180	480	180	11800	12600	1.067797
2	2000	0	1320	1140	1540	740	1590	900	500	600	340	820	240	220	680	480	13100	13008	0.992977
3	1780	1340	0	1500	1580	680	2420	720	360	1320	420	900	220	380	820	580	15020	14772	0.963485
4	1820	1160	1380	0	1560	1020	2240	1060	680	1460	800	740	280	280	500	420	15400	14184	0.921039
5	2300	1520	1620	1700	0	1260	2240	1260	680	1420	1420	1980	380	340	1280	520	19900	20095	1.009799
6	1500	980	920	1240	1420	0	1620	640	740	1180	860	1040	600	540	980	740	15200	15343	1.009408
7	2900	1500	2420	2300	2220	1420	0	720	500	980	1020	960	520	420	1340	480	19800	20863	1.053687
8	1980	1260	1060	1340	1580	820	980	0	900	1540	700	920	520	480	1000	720	15800	14511	0.918418
9	1120	760	580	960	940	780	740	920	0	580	560	680	460	440	600	400	10520	10951	1.04097
10	1200	620	1240	1480	1400	920	920	1200	260	0	720	900	360	360	580	440	12600	11196	0.888571
11	940	480	560	920	1480	680	1020	500	400	900	0	680	340	340	660	400	10300	11392	1.106019
12	1040	860	900	880	1980	860	900	760	460	1020	660	0	500	420	920	360	12520	11294	0.902077
13	920	540	460	500	620	640	680	500	440	620	560	680	0	600	820	320	9000	10020	1.113333
14	1060	560	560	500	680	600	660	480	480	700	600	720	600	0	720	380	9400	9955	1.059043
15	1060	660	860	500	1340	780	1300	800	420	640	640	880	560	400	0	280	11220	11751	1.047326
16	1140	800	880	860	900	840	760	740	500	800	680	660	440	400	580	0	10980	11490	1.046448
d	22760	14480	15920	17400	20880	12660	20340	12480	7500	14320	10240	13020	6120	5800	11940	6700	212560	213425	1.004069
E	12600	13008	14772	14184	20095	15343	20863	14511	10951	11196	11392	11294	10020	9955	11751	11490	213425		
E (O/d)	0.594	0.896	0.928	0.815	0.962	1.212	1.026	1.163	1.46	0.782	1.113	0.867	1.637	1.72	0.984	1.715	1.00407		

Hasil Proses Iterasi I

Tabel B.2 Hasil Iterasi 1

ITERASI 1																				
O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	I	T	E(I/T)	
1	0	1431	1209	1367	1751	662	2435	1159	192.2	590	277.6	491.2	106.6	192	512.5	192.2	12600	12600	1	
2	1996	0	1311	1132	1529	734.8	1569	893.7	496.5	595.3	337.6	814.2	238.3	218	675.2	476.6	13008	13008	1	
3	1752	1318	0	1475	1554	668.8	2380	708.1	354.1	1298	413.1	885.1	216.4	374	806.5	570.4	14772	14772	1	
4	1676	1068	1271	0	1437	939.5	2063	976.9	626.3	1345	736.8	681.6	257.9	290	460.5	386.8	14184	14184	1	
5	2323	1535	1636	1717	0	1272	2262	1272	686.7	1434	1434	1999	383.7	343	1272	525.1	20095	20095	1	
6	1514	989.2	928.7	1252	1433	0	1635	647.9	747	1191	868.1	1050	605.6	545	989.2	747	15343	15343	1	
7	2536	1686	2590	2423	2339	1496	0	758.7	526.8	1033	1075	1012	547.9	443	1412	505.8	20863	20863	1	
8	1618	1157	973.5	1231	1451	753.1	900	0	826.6	1414	642.9	844.9	477.6	441	918.4	601.3	14511	14511	1	
9	1166	791.1	603.8	999.3	978.5	812	770.3	957.7	0	603.8	582.9	707.9	478.8	458	624.6	416.4	10951	10951	1	
10	1066	550.9	1102	1315	1424	817.5	1175.5	1066	231	0	639.8	799.7	319.9	320	515.4	391	11196	11196	1	
11	1040	530.9	619.4	1018	1637	752.1	1128	553	442.4	995.4	0	752.1	376	376	730	442.4	11392	11392	1	
12	938	2	775.8	811.9	793.8	1786	775.8	811.9	685.6	415	920.1	595.4	0	451	379	829.9	324.7	11294	11294	1
13	1024	601.2	512.1	668	690.3	712.5	757.1	556.7	489.9	690.3	623.5	757.1	0	668	912.8	354.4	10020	10020	1	
14	1123	593.1	593.1	635.4	720.1	635.4	699	508.3	508.3	741.3	635.4	762.5	635.4	0	762.5	432.4	9955	9955	1	
15	1110	691.2	900.7	629.4	1403	816.9	1762	837.9	439.9	670.3	670.3	921.6	586.5	419	0	293.3	11751	11751	1	
16	1193	837.2	920.9	899.9	941.8	879	774.4	879	523.2	837.2	711.6	690.7	460.4	419	606.9	0	11490	11490	1	
d	22784	14556	15972	17554	20966	12726	20385	12550	7506	14367	10244	13169	6142	5852	12029	6692	213425	213425		
E	12600	13008	14772	14184	20095	15343	20863	14511	10951	11196	11392	11294	10020	9955	11751	11490	213425			
E (O/d)	0.553	0.894	0.925	0.808	0.962	1.205	1.023	1.156	1.459	0.779	1.112	0.858	1.631	1.71	0.977	1.717				

Hasil Proses Iterasi II

Tabel B.3 Hasil Iterasi II

ITERASI 2																			
O/D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	I	T	E(I/T)
1	0	1279	1146	1194	1694	796.1	2492	1330	280.4	466	308.8	421.2	174.2	327	500.7	330	12644.2	12600	0.996506
2	1098	0	1212	914.7	1471	885.8	1606	1033	724.4	464.3	375.5	698.3	388.8	372	659.6	818.4	12721.4	13008	1.022528
3	968.1	1178	0	1152	1494	806.2	2436	818.8	516.6	1012	459.4	759.1	353	636	787.8	979.5	14395.7	14772	1.026137
4	927.1	954.8	1176	0	1382	1132	2112	1129	913.8	1048	819.4	584.5	420.7	439	449.9	664.2	14151.1	14184	1.002323
5	1361.1	1372	1614	1703	0	1167	2262	1272	686.7	1434	1434	1999	383.7	343	1272	525.1	20095	20095	1

Hasil Proses Iterasi IV

Tabel B.5 Hasil Iterasi IV

		ITERASI J																	
O / D	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	i	I	E (1/I)
0	1	0.1272	1138	1097	1661	794.3	2496	1332	279.8	464.2	307.8	419.4	714.4	327.4961	329.2	12650.7	12600		0.999942
2	1118	0	1.236	932.1	1506	904.6	1456	1059	74.1	713.3	399.3	382	673.4	837.7	13006.1	13008	1.000148		
3	989.3	1207	0	1.219	1536	826.2	2503	842.2	530.8	1038	471.6	778.2	363.8	655	807.1	1006	14772.6	14772	0.999945
4	925.3	955.5	1175	0	1.387	1134	2119	1134	917.2	1050	821.8	585.3	423.6	442	450.2	666.5	14186.1	14184	0.999855
5	1307	1400	1542	1413	0	1.566	2369	1507	1025	1142	1631	1751	642.8	600	1266	922.6	20087.5	20089	1.000373
6	816.3	864.1	838.3	986.8	1352	0	1.641	962.2	1068	908.4	945.7	880.5	97.1	912	944.5	1257	15348.2	15343	0.999661
7	1758	1571	2456	2308	2353	1882	0	918.5	804	840.2	1249	905.2	937.9	790	1438	908	20589.5	20863	1.000649
8	1004	1036	900.3	994	1042	909.4	925.1	0	1.211	1015	717.5	726	784.9	756	698.4	1140	14059.1	14511	1.000113
9	630.9	693.6	547.1	790.8	925.1	960.6	75.7	1091	0	462.2	632.4	595.9	79.1	769	598.6	703.3	10593.3	10951	0.999786
10	577.2	483.2	998.6	1041	1178	967.5	823.5	1215	331.8	0	699.7	673.5	515.3	537	694.1	660.6	11195.8	11196	1.000014
11	566.6	456.8	565.2	810.9	596	896.1	1144	634.3	639.7	767.4	0	637.7	609	636	704.6	752.5	11394.2	11392	0.99981
12	504.2	673.1	728	621.6	1673	908.3	809.1	772.7	589.6	697	644.2	0	718.8	630	787.1	542.8	11297.4	11294	0.999703
13	549.1	522.2	459.7	523.6	642.7	835.1	728.1	626.1	696.7	523.5	675.3	631.4	0	1111	686.7	596.1	10021.2	10020	0.999878
14	607.7	520.2	537.6	503	661.8	752.1	704.2	579.2	730.1	567.7	695	642.2	1024	0	731	680	9955.44	9955	0.999556
15	591.7	596.9	603.8	489.7	1308	951.9	1350	599.9	622	505.4	721.8	764.2	932.0	693	0	487.8	11756.7	11751	0.999512
16	546.5	745.2	847.1	723	905	1056	813.1	895.5	762.7	650.7	789.9	590.3	752.8	714	590.6	0	11490.9	11490	0.999992
d	12600	13006	14772	14184	20095	15343	20663	14511	10951	11196	11362	11294	10020	9955	11751	11490	213425	213425	1
d	12600	13006	14772	14184	20095	15343	20663	14511	10951	11196	11362	11294	10020	9955	11751	11490	213425	213425	1
E (D/d)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

LAMPIRAN C

Lampiran C berisi data rute lama dan rute baru dari trayek A sampai trayek B beserta waktu perjalanan, dan panjang masing-masing rutenya.

Hasil Proses Mencari Rute Baru Trayek A

```
Trayek A
--- Arah Keberangkatan ---
Rute Lama :
1001-1101-1102-0502-0104-0401-0212-0113-0112-0111-0205-0204-0202-0304-0203-0206-
0205-0701-0706-0705-0702-0703-0707
Waktu Perjalanan : 14.7546731023216
Panjang Rute : 7920
Rute Baru :
1001-1101-1102-0502-0104-0401-0212-0113-0112-0111-0205-0204-0202-0304-0203-0206-
0205-0701-0702-0707
Waktu Perjalanan : 12.824661794972
Panjang Rute : 7570
--- Arah Pulang ---
Rute Lama :
0707-0703-0702-0705-0706-0701-0305-0206-0303-0301-0211-0210-0112-0113-0212-0401-0104-
0502-1102-1101-1001
Waktu Perjalanan : 14.0288607009608
Panjang Rute : 7257
Rute Baru :
0707-0703-0702-0701-0205-0206-0303-0301-0211-0210-0112-0113-0212-0401-0104-0502-1102-
1101-1001
Waktu Perjalanan : 12.107089226222
Panjang Rute : 6907
```

Gambar C.1. Rute Baru Trayek A

Hasil Proses Mencari Rute Baru Trayek B

```
Trayek B
--- Arah Keberangkatan ---
Rute Lama :
1001-1101-1102-0501-0502-1201-1301-1501-0604-0601-0602-0207-0208-0209-0202-0201
Waktu Perjalanan : 16.610455808806
Panjang Rute : 8760
Rute Baru :
1001-1101-1102-0501-0502-1201-1301-1501-0604-0601-0602-0207-0208-0209-0202-0201
Waktu Perjalanan : 16.610455808806
Panjang Rute : 8760
--- Arah Pulang ---
Rute Lama :
0201-0213-0204-0205-0111-0112-0113-0212-0401-0304-0503-1002-1001
Waktu Perjalanan : 8.4231112491089
Panjang Rute : 4613
Rute Baru :
0201-0213-0204-0205-0111-0112-0113-0212-0401-0304-0503-1002-1001
Waktu Perjalanan : 8.4231112491089
Panjang Rute : 4613
```

Gambar C.2. Rute Baru Trayek B

Hasil Proses Mencari Rute Baru Trayek C

Trayek C



Hasil Proses Mencari Rute Baru Trayek D

```
Trayek D
--- Arah Keberangkatan ---
Rute Lama :
1001-1002-0503-0104-0401-0103-0101-0106-0107-0501-0502-0604-1501-1201
Waktu Perjalanan : 14.6268689428393
Panjang Rute : 7807
Rute Baru :
1001-1002-0503-0104-0401-0103-0101-0106-0107-0501-0502-0604-1501-1201
Waktu Perjalanan : 14.6268689428393
Panjang Rute : 7807
--- Arah Pulang ---
Rute Lama :
1201-1201-1202-0501-1103-1101-1001
Waktu Perjalanan : 7.16583753945195
Panjang Rute : 4630
Rute Baru :
1201-1201-1202-0501-1103-1101-1001
Waktu Perjalanan : 7.16583753945195
Panjang Rute : 4630
```

Gambar C.4. Rute Baru Trayek D

Hasil Proses Mencari Rute Baru Trayek E

```
Trayek E
--- Arah Keberangkatan ---
Rute Lama :
0804-0801-0802-0402-0212-0112-0112-0111-0205-0206-0110-0602-0603-0208-1604-1601
Waktu Perjalanan : 11.7041846505635
Panjang Rute : 6347
Rute Baru :
0804-0801-0802-0402-0212-0112-0112-0111-0205-0206-0110-0602-0603-0208-1604-1601
Waktu Perjalanan : 11.7041846505635
Panjang Rute : 6347
--- Arah Pulang ---
Rute Lama :
1601-1604-0208-0209-0202-0201-0213-0204-0205-0210-0211-0212-0402-0802-0801-0804
Waktu Perjalanan : 13.410464210297
Panjang Rute : 6500
Rute Baru :
1601-1604-0208-0209-0202-0201-0213-0204-0205-0210-0211-0212-0402-0802-0801-0804
Waktu Perjalanan : 13.410464210297
Panjang Rute : 6500
```

Gambar C.5. Rute Baru Trayek E

Hasil Proses Mencari Rute Baru Trayek F

```
Trayek F
--- Arah Keberangkatan ---
Rute Lama :
1001-0901-0803-0801-0802-0402-0212-0211-0301-0303-0304
Waktu Perjalanan : 10.144053995887
Panjang Rute : 6310
Rute Baru :
```


Hasil Proses Mencari Rute Baru Trayek G

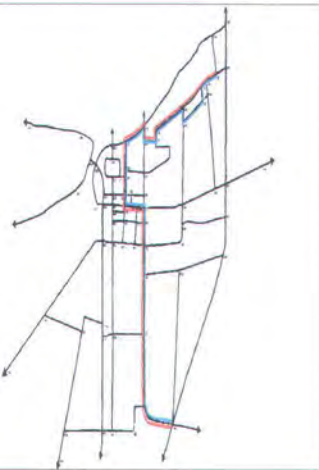
```
Trayek G
--- Arah Keberangkatan ---
Rute Lama :
1001-0901-0803-1004-1003-0104-0105-0106-0101-0102-0111-0112-0210-0211-0301-0303-0304
Waktu Perjalanan : 15.6382291145433
Panjang Rute : 7255
Rute Baru :
1001-0901-0803-0801-0802-0402-0212-0401-0104-0105-0106-0101-0102-0111-0112-0113-0212-
0211-0301-0303-0304
Waktu Perjalanan : 14.6291624231385
Panjang Rute : 8369
--- Arah Pulang ---
Rute Lama :
0304-0303-0301-0211-0210-0112-0113-0212-0401-0104-1003-1004-0803-0901-1001
Waktu Perjalanan : 14.5117997726335
Panjang Rute : 6876
Rute Baru :
0304-0303-0301-0211-0210-0112-0113-0212-0401-0104-1003-1004-0803-0901-1001
Waktu Perjalanan : 14.5117997726335
Panjang Rute : 6876
```

Gambar C.7. Rute Baru Trayek G

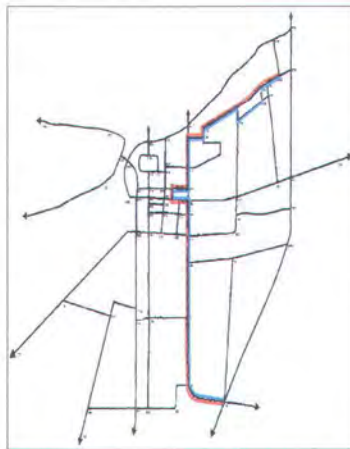
LAMPIRAN D

Lampiran D berisi gambar peta masing-masing trayek, antara rute lama dan rute barunya untuk tiap arah berangkat dan arah pulang. Dengan warna merah adalah rute baru dan warna biru adalah rute lama.

a Trayek A (Arah Berangkat): Peta Trayek A (Arah Pulang):

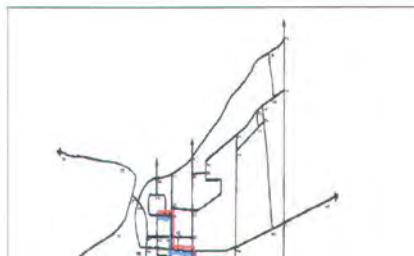
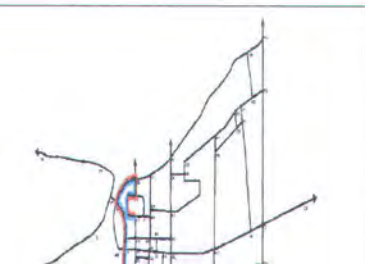


Gambar D.1 Trayek A Berangkat

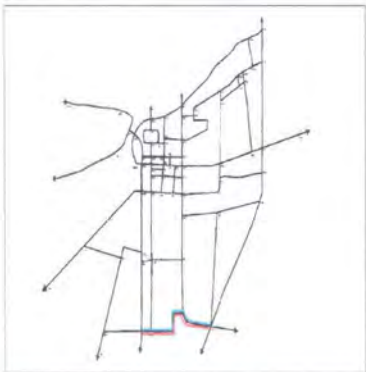


Gambar D.2 Trayek A Pulang

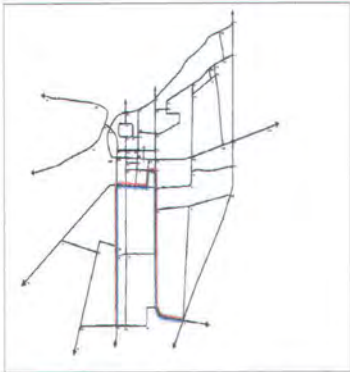
a Trayek B (Arah Berangkat): Peta Trayek B (Arah Pulang):



Peta Trayek C (Arah Berangkat):

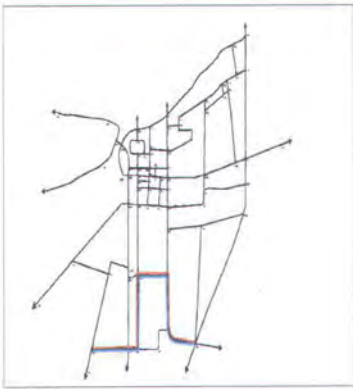
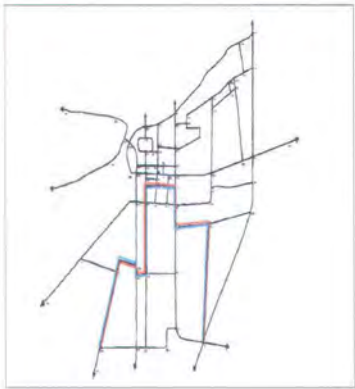


Peta Trayek C (Arah Pulang):



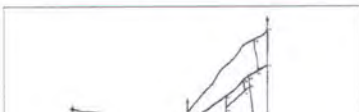
Gambar D.5 Trayek C Berangkat
Peta Trayek D (Arah Berangkat):

Gambar D.6 Trayek C Pulang
Peta Trayek D (Arah Pulang):

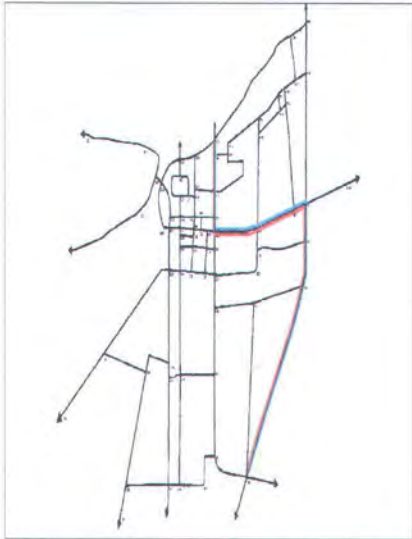


Gambar D.7 Trayek D Berangkat
Peta Trayek E (Arah Berangkat):

Gambar D.8 Trayek D Pulang
Peta Trayek E (Arah Pulang):

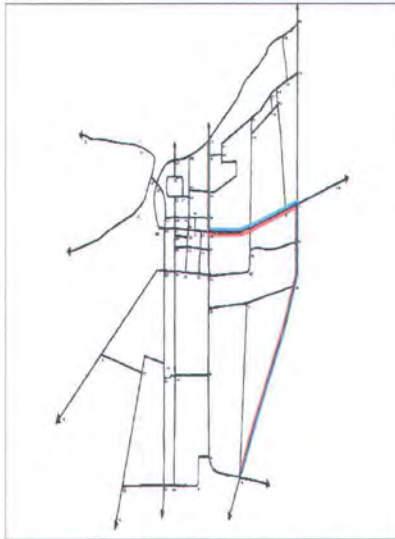


Peta Trayek F (Arah Berangkat):



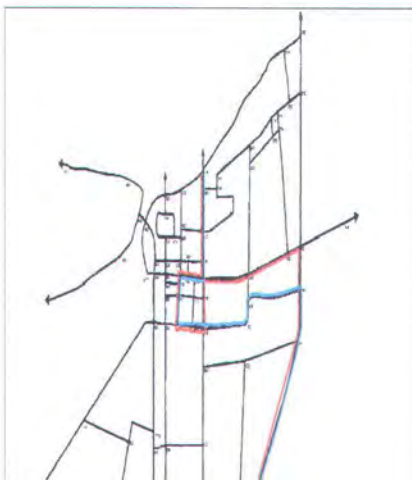
Gambar D.11 Trayek F Berangkat

Peta Trayek F (Arah Pulang):

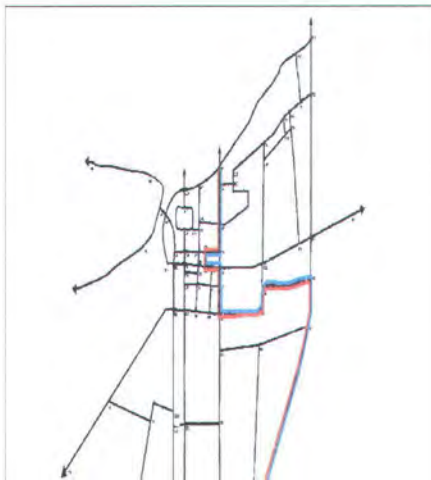


Gambar D.12 Trayek F Pulang

Peta Trayek G (Arah Berangkat):



Peta Trayek G (Arah Pulang):



LAMPIRAN E

Lampiran E berisi data hasil keluaran SPSS 13.0, untuk menguji persamaan regresi yang dijelaskan pada bab 6.

Variabel Yang Digunakan Dalam Persamaan Regresi:

Variables Entered/Removed ^a			
Model	Variables Entered	Variables Removed	Method
1	Total_Penduduk		Stepwise (Criteria: Probability <= .050, Probability >= .100)

^a. Dependent Variable: Total_Perjalanan

Gambar E.1. Hasil SPSS (1)

Nilai Koefisien Determinasi/ R^2 :

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.964 ^a	.930	.925	915.03599

^a. Predictors: (Constant), Total_Penduduk

Gambar E.2. Hasil SPSS (2)

Nilai F untuk Uji-F :

Coefficients ^a					
Model		Unstandardized Coefficients		Standardized Coefficients	Sig.
		B	Std. Error	Beta	
1	(Constant)	7505.410	482.661		.000
	Total_Penduduk	.809	.059	.964	.000

^a. Dependent Variable: Total_Perjalanan

Gambar E.3. Hasil SPSS (3)

Nilai Konstanta Dan Koefisien Persamaan Regresi Serta Nilai t Untuk Uji-t :

LAMPIRAN E

Lampiran E berisi data hasil keluaran SPSS 13.0, untuk menguji persamaan regresi yang dijelaskan pada bab 6.

Variabel Yang Digunakan Dalam Persamaan Regresi:

Variables Entered/Removed ^a			
Model	Variables Entered	Variables Removed	Method
1	Total_Penduduk		Stepwise (Criteria: Probabilit y-of- F-to-enter ≤ .050, Probabilit y-of- F-to-remo ve ≥ .100)

a. Dependent Variable: Total_Perjalanan

Gambar E.1. Hasil SPSS (1)

Nilai Koefisien Determinasi/ R^2 :

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.964 ^a	.930	.925	915.03599

a. Predictors: (Constant), Total_Penduduk

Gambar E.2. Hasil SPSS (2)

Nilai F untuk Uji-F :

Coefficients ^a					
Model		Unstandardized Coefficients		Standardized Coefficients	
		B	Std. Error	Beta	
1	(Constant)	7505.410	482.661		15.550
	Total_Penduduk	.809	.059	.964	13.598

a. Dependent Variable: Total_Perjalanan

Gambar E.3. Hasil SPSS (3)

Nilai Konstanta Dan Koefisien Persamaan Regresi Serta Nilai t Untuk Uji-t :

LAMPIRAN F

Lampiran F berisi contoh jaringan jalan yang digunakan dalam uji algoritma dijkstra

Contoh Jaringan Jalan (1) :



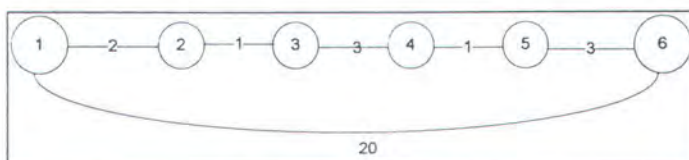
Gambar F.1. Jaringan Jalan (1)

Hasil Proses Dijkstra (1):

Node Awal
1
Node Akhir
9
Route
1 - 3 - 2 - 6 - 9

Gambar F.2. Hasil Dijkstra (1)

Contoh Jaringan Jalan (2) :



Gambar F.3. Jaringan Jalan (2)

LAMPIRAN G

Lampiran G berisi contoh data yang digunakan untuk proses aplikasi perencanaan trayek angkutan kota yang disimpan dalam sebuah basis-data perencanaan Trayek', yang dapat dilihat pada tabel-tabel di bawah ini.

Tabel Asal Tujuan Perjalanan sebagai berikut:

Tabel G.1. Tabel Asal Tujuan Perjalanan

Tahun Pendataan	Zona Asal	Zona Tujuan	Jumlah Perjalanan	Jumlah Perjalanan
2005	1	1	0	0
2005	1	2	67	14
2005	1	3	58	10
2005	1	4	64	6
2005	1	5	82	12
2005	1	6	31	3
2005	1	7	114	40
2005	1	8	54	9
2005	1	9	9	0
2005	1	10	28	2
2005	1	11	13	1
2005	1	12	23	2
2005	1	13	5	0
2005	1	14	9	0
2005	1	15	24	2
2005	1	16	9	0
2005	2	1	100	31
2005	2	2	0	0
2005	2	3	66	13
2005	2	4	57	4
2005	2	5	77	18
2005	2	6	37	4
2005	2	7	79	19
2005	2	8	45	6
2005	2	9	25	2
2005	2	10	30	3
2005	2	11	17	1

Tabel Jalan sebagai berikut:

Tabel G.2. Tabel Jalan

ID Jalan	Nama Jalan
1	BY PASS
2	GAJAH MADA
3	PAHLAWAN
4	JAYA NEGARA
5	MOJOPAHIT
6	HAYAM WURUK
7	BRAWIJAYA
8	RA BASUMI
9	JEND A YANI
10	...

Tabel Kelurahan sebagai berikut:

Tabel G.3. Tabel Kelurahan

	id_kelurahan	nama_kelurahan
▶	7	Sentanan
	8	Jagalan
	9	Purwatengah
	10	Purwatengah
	11	Kauman
	12	Mentikan
	13	Magersari
	14	Gedongan
	15	Balongsari
	16	Kranggan
	17	Miji
	18	Mentikan
	19	Miji

Tabel Kependudukan sebagai berikut:

Tabel G.4. Tabel Kependudukan

	ID_Kependudukan	ID_Komponen_Kep	Tahun	Nilai
▶	1	1	1998	107123
	2	1	1999	108027
	3	1	2000	108858
	4	1	2001	109911
	5	1	2002	111249
	6	1	2003	112547
	7	1	2004	113275
	8	2	2000	1882363162
	9	2	2001	2182655381
	10	2	2002	2489200083
	11	2	2003	2826500356
	12	2	2004	3215921301
	13	3	2002	23312
	14	3	2003	26562
	15	3	2004	30641
*				

Tabel Komponen Kependudukan sebagai berikut:

Tabel G.5. Tabel Komponen Kependudukan

	ID_Komponen_Kep	Nama_Komponen_Kep
▶	1	Jumlah Penduduk
	2	Jumlah Pendapatar
	3	Kepemilikan Kendar
*		

Tabel Node sebagai berikut:

Tabel G.7. Tabel Node

	ID_Node	Nama_Node	ID_Zona	Node_Pusat
▶ 1		0101	1	0
	2	0102	1	0
	3	0103	1	0
	4	0104	1	0
	5	0105	1	0
	6	0106	1	0
	7	0107	1	1
	8	0108	1	0
	9	0109	1	0
	10	0110	1	0
	11	0111	1	0
	12	0112	1	0
	13	0113	1	0
	14	0201	2	1
	15	0202	2	0
	16	0203	2	0

Tabel Rute sebagai berikut:

Tabel G.8. Tabel Rute

	ID_Trayek	ID_Link	No_Urut	Arah_Berangkat
▶ 1		157	1	1
	1	168	2	1
	1	172	3	1
	1	100	4	1
	1	9	5	1
	1	89	6	1
	1	64	7	1
	1	34	8	1
	1	31	9	1
	1	30	10	1
	1	47	11	1
	1	44	12	1
	1	43	13	1
	1	75	14	1
	1	80	15	1
	1	78	16	1
	1	86	17	1

Tabel Sampel HI sebagai berikut:

Tabel G.9. Tabel Sampel HI

	ID_Zona	Tahun_Pendataan	Jumlah_Keluarga	Jumlah_Seluruh_Keluarga	Jumlah_Rumah_Ko
▶ 1		2005	69	1380	0
	2	2005	72	1445	0

Tabel Total Data Keluarga sebagai berikut:

Tabel G.10. Tabel Total Data Keluarga

ID Total Data Kel	ID_Zona	Tahun Pendataan	Total Anggota_Kel	Total Pendapatan	Total Kendaraan
1	1	2005	312	9065249	89
2	2	2005	337	9871387	108
3	3	2005	445	13014817	141
4	4	2005	409	11967852	111
5	5	2005	771	21182667	199
6	6	2005	480	14771590	155
7	7	2005	818	24642976	267
8	8	2005	429	17187468	121
9	9	2005	211	7658310	61
10	10	2005	226	8030959	64
11	11	2005	238	7518884	74
12	12	2005	232	8304741	70
13	13	2005	154	5726621	48
14	14	2005	150	5653105	39
15	15	2005	260	7752106	75
16	16	2005	244	6882592	61
*					

Tabel Trayek sebagai berikut:

Tabel G.11. Tabel Trayek

ID Trayek	Nama Trayek	Tarif
1	A	2000
2	B	2000
3	C	2000
4	D	2000
5	E	2000
6	F	2000
7	G	2000
*		

Tabel Wilayah Zona sebagai berikut:

Tabel G.12. Tabel Wilayah Zona

id_zona	id_kelurahan
1	7
2	8
3	7
4	9
5	8
6	9
7	8
8	7
9	7
*	

Tabel Zona sebagai berikut:

